PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Shiri KADAMBI et al. Group Art Unit: 2666

Serial No.: 09/635,232 Examiner: GholamReza, Zahedian Taknaki

Filed: August 9, 2000 Atty. Docket No.: 58268.00021

For: A METHOD FOR SENDING PACKETS BETWEEN TRUNK PORTS OF NETWORK SWITCHES

DECLARATION UNDER 37 CFR §1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
Sir:

I, W. Douglas Carothers, Jr., a citizen of the United States and residing in Cupertino, California, hereby declare and state:

1. I was one of the attorneys that assisted the inventors of the above-cited application, Shiri KADAMBI and Shekhar AMBE, in preparing invention disclosures and provisional patent applications, Serial No. 60/092,220, filed July 8, 1998, and Serial No. 60/095,972, filed August 10, 1998, on which the present application claims priority under 35 U.S.C. §119(e).

2. Prior to June 29, 1998, I attended a meeting with the inventors at the place of their former employer, Maverick Networks, in San Jose, California, where the invention disclosed in the above mentioned provisional applications was discussed. Maverick Networks was acquired by Broadcom Corporation in 1999.

3. A copy of the invention disclosure given to me prior to that meeting, and my notes from that meeting made on and included with the invention disclosure, are enclosed with this declaration. The invention disclosure and my notes were used in preparation of the provisional applications. Most of the subject matter of the invention disclosure and my notes were incorporated into one or both of the above-cited filed provisional applications.

4. Given the completeness of the invention disclosure and the discussions carried out at the above-cited meeting with the inventors, I can state that the inventions disclosed in provisional patent applications Serial Nos. 60/092,220 and 60/095,972 were invented prior to June 29, 1998.

4. I hereby declare that all statements made herein of my own knowledge are true, and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine and/or imprisonment under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing therefrom.

Date: APRIL 2, 2004

W. Douglas Carothers, Jr.

(1)

MP1002

Claims to be Written

1.) Shared Memory
- on chip + off chip memory
- utilization by a small/all number of I/O modules

2.) a share of Table data + status among EPIC

3.) Packet → cell proprietary design to facilitate memory use & sharing among the I/O modules, particularly EPIC units

4.) Rerouting algorithm for packet assembly in GBP using external DRAM + on-chip SRAM.
- Using the packet FIFO to allocate pointers to both off-chip + on-chip memory.

5.) Tables in silicon (either off-chip + are loaded by CPU. Table management - self-management on the chip. & CPU is using sideband 5 channel simultaneously (with

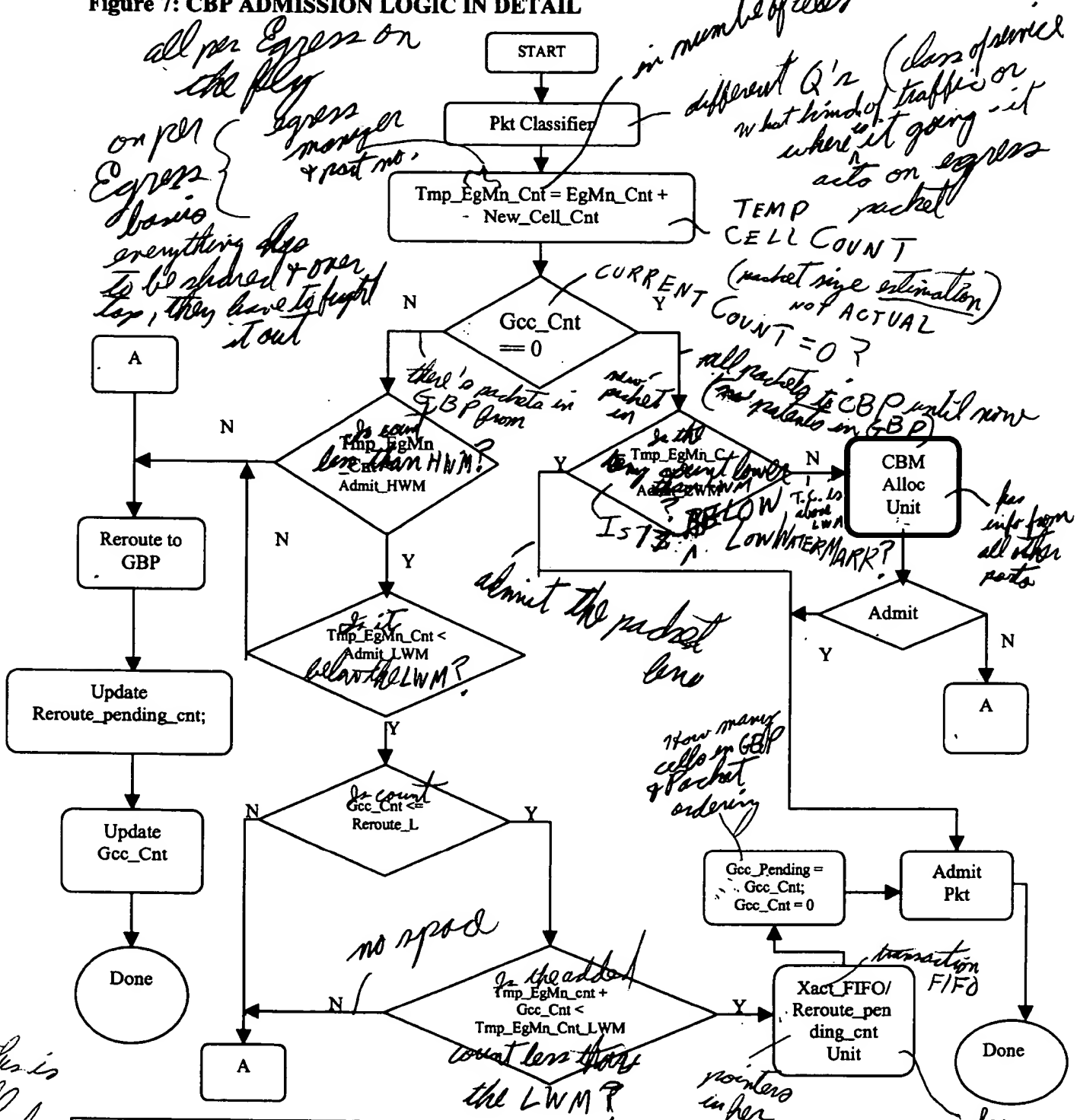
(2)

the chip using the C channel to transfer cells & P channel to transfer the commands.

- CPU can be using the serial port to transmit packets via CMVE to C-P channels while communicating on the S channel to the

6.) all of the chips use off & on chip memory to maximize performance & memory costs to end user

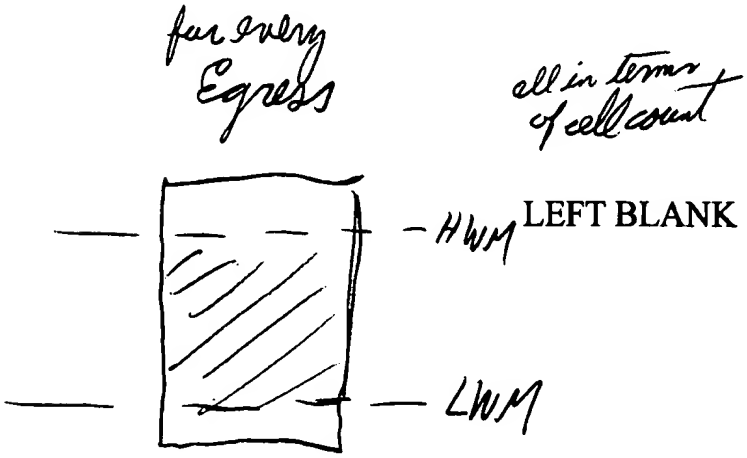
Figure 7: CBP ADMISSION LOGIC IN DETAIL



Witnessed and Understood By:	Date:
Submitter(s) Signature(s):	Date(s):

GCC_Cnt = Current count of cells in GBP.
Admit_LWM = Enables reception of new packets into the CBP if the total number of cells in the Egn (Egress n) is below this cell count. This being true by itself is not sufficient enough to allow the packet into the CBP.
Admit_HWM = Disable reception of new packets above this count in the CBP.
GCC_Pending_Cnt = Temp register to hold GCC_Cnt. Used during the re-admission process of new packets (from Ingress) directly into the CBP.
Tmp_EgMn_Cnt = Egress Manager n , current cell count.
Reroute_L = Programmable register . Enables admission of new packets into the CBP only if GCC_Cnt < Reroute_L. This being true by itself is not sufficient enough to allow the packet into the CBP.
Reroute_pending_cnt = Number of packets rerouted and still waiting for GPID assignment.

The Egress Manager Scheduler (refer to figure 11) is part of the Egress Manager and is discussed later. The Reclaim unit is used in cases where the packet gets dropped due to the P (Purge) bit getting set. In this case the Reclaim unit cleans up the memory by flushing out the dirty cells of the packet and writing back the cell pointers into the FAP. In order for this to occur the first cell pointer needs to be stored till the whole packet gets written into the memory. The Reclaim unit is illustrated in Figure 9.



Witnessed and Understood By:	Date:
Submitter(s) Signature(s):	Date(s):

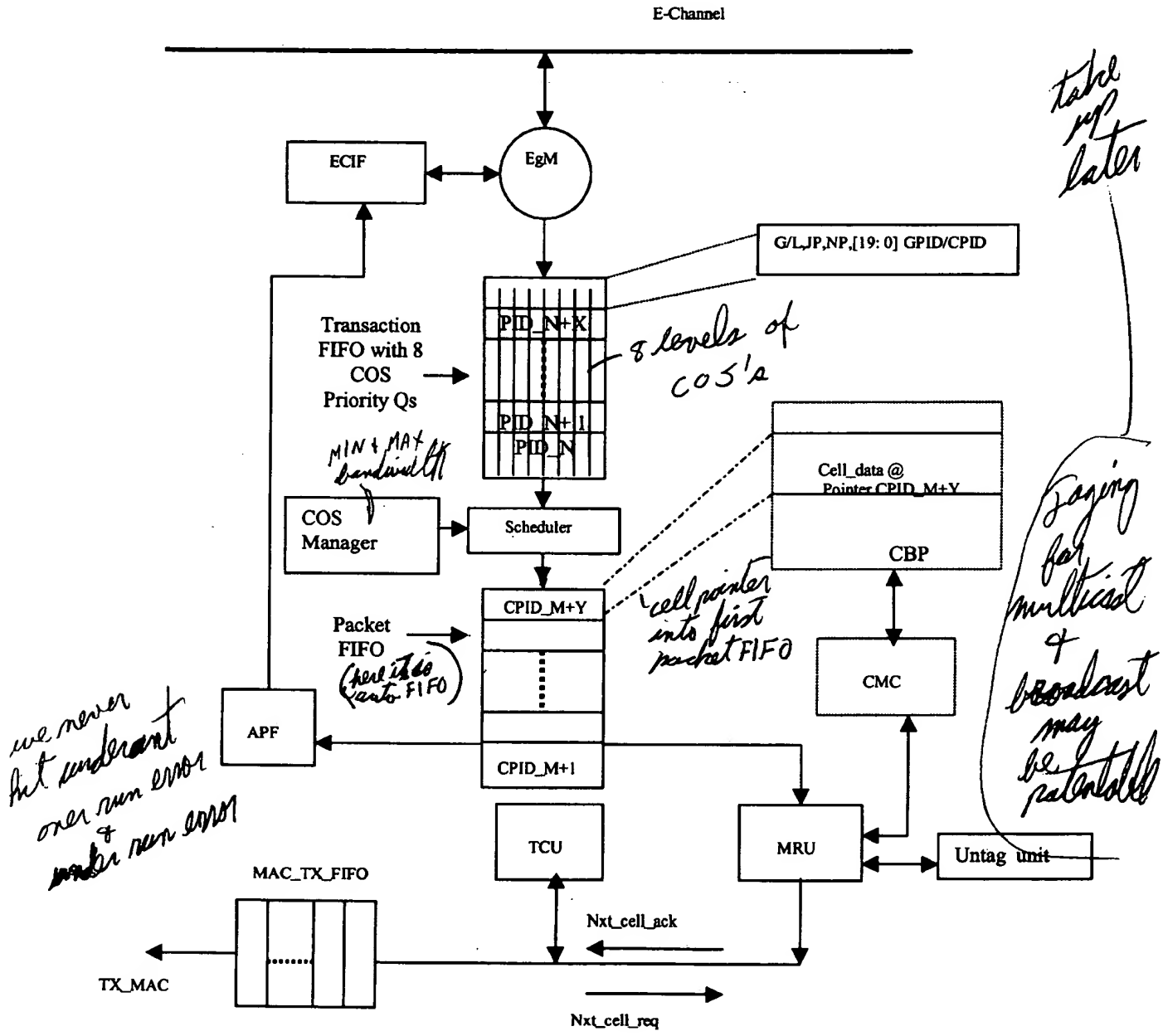


Figure 10: Egress Manager (EgM).

Witnessed and Understood By:	Date:
Submitter(s) Signature(s):	Date(s):

To: Patent Operations		
1	Proposal Submitted By (Use Legal Name - First , Middle Last) **Shiri Kadambi**	Organization, Building & Extension **Maverick Networks**
2	Proposal Submitted By (Use Legal Name - First , Middle Last) **Shekhar Ambe**	Organization, Building & Extension **Maverick Networks**
*3	Proposal Submitted By (Use Legal Name - First , Middle Last)	Organization, Building & Extension
	* If space for additional submitters is required, please use an additional form. Each named submitter must also sign and date each page of the Disclosure.	

Descriptive Title of Invention Disclosure:
High Performance Self Balancing Low Cost Network Switching Architecture Based On Distributed Hierarchical Shared Memory

Indicate any Program or Product Name and any Expected Date of Sale or Shipment:	Identify Related IDs and Technical Keywords for Searching:
Product Name : Orion Date of Shipment : Q2 of 1999	Distributed Hierarchical Shared Memory, SOC, Maverick Networks.

If This Disclosure is Funded Under a Contract, Provide the Contract Number, Customer Name and Customer Program Name: No

If a Prototype is to be Delivered, Based Upon the Disclosure, Indicate the Expected Date of Sale or Shipment of the Prototype: No

If any Portion of This Disclosure Has Been Previously Published or Presented in an Outside Presentation, Indicate the Date of Disclosure and the Portion Disclosed: No

Names of Others Known to Have Worked on the Same Product or Technology and Citation of Known Published Works: **

** Attach Copies of any papers or patents if submitter(s) already has copies.

Description of Concept and Embodiments:
(The preferred form is to include background information on invention and existing problems, followed by description of the proposed invention with drawings and a discussion as how the new concept improves over present technology. It is permissible to attach copies of materials such as lab notebook pages, memos, drawings, etc. and to refer to such items in the body of the description below, providing all such materials, including this Invention Disclosure, are signed and dated by each named submitter(s) and the witness.)

Attached

68

Witnessed and Understood By:	Date:
Submitter(s) Signature(s):	Date(s):

INVENTION DISCLOSURE FOR

HIGH PERFORMANCE, SELF BALANCING, LOW COST NETWORK SWITCHING ARCHITECTURE BASED ON DISTRIBUTED HIERARCHICAL SHARED MEMORY

Submitter(s): Shiri Kadambi and Shekhar Ambe

Summary

A new Switch Architecture with distributed hierarchical shared memory providing a very low cost switch solution for Fast Ethernet and Gigabit Ethernet Switches. The architecture includes a way to achieve dynamic rerouting of Packets to external Memory (also called Global Memory Pool – GBP) or Memory on Chip (also called Common Buffer Pool - CBP), by employing unique algorithms for the packet assembly by CBP / GBP Admission Logic.

Introduction

The semiconductor industry has been characterized as following "Moore's Law" where the performance / price doubles every 18 months. The same trend is happening in networking industry where the bandwidth / price doubles less than 18 months. To keep up with this trend, need arises for an innovative Switch Fabric Architecture. The Switch Fabric Architecture should be 1) highly scalable, 2) support layer 2 and layer 3 switching, 3) rich in features 4) support extensive Filtering Mechanism, 5) offers highly integrated solution and 6) highly cost effective.

The Maverick's Switch On Chip (SOC) Architecture is designed with all the above objectives in mind.

Background

The Switch Fabric Solutions currently available in the market is a multi Chip Solution (more than 3 chips and in many cases even 9 chips). The multiple Chip Solution needs external glue logic circuitry, which shoots up the cost and lowers performance. The Maverick's Switch On Chip (SOC) Architecture is a highly integrated Switch Fabric Solution. It provides single Chip solution for a 24 100MB Ports and 2 Gigabit Port Switch, thus bringing the cost down.

Presently, most of the switches available in the market use very expensive SRAMS to achieve higher throughput. In most of the conventional Shared Memory Switch Architecture, the Packet Memory, where the incoming packets are stored, comprises of typically 4MB to 8MB SRAM, which brings up the Switch cost still further. SOC's unique distributed hierarchical shared memory architecture uses On Chip Memory and uses DRAMs for its Global memory, thus bringing the cost down.

The Maverick's Switch On Chip (SOC) is architectured to achieve high performance using standard DRAMS. SOC achieves this through an innovative Buffer Management Scheme, coupled with Self Balancing Traffic Flow dependent rerouting Algorithm.

The SOC Architecture is very rich in features. Most of the Switch Fabrics available in the market supports only layer 2 switching. SOC Architecture supports both Layer 2 and Layer 3 Switching. It uses innovative technique for doing the Layer 3 Route Lookup. The Layer 3 Switching Logic uses Route Cache for End Stations connected directly to one of the L3 Interfaces of the switch and Default Router Table for the Stations that are not directly connected to one of the L3 interfaces. One can support large number of stations which requires L3 switching by using relatively smaller Layer 3 Route Tables.

Most of the Switch Fabric available in the market supports very primitive Filtering Mechanism. Many switch vendors use very powerful CPU to implement the Filtering packet by packet basis. Most of the Switches available in the market today support filters that operate on layer 2 to layer 4 of the packets. At this point no Switch vendor has the Filtering capability that enables the Switch Application to set the filter on any field from Layer 2 to Layer 7.

SOC Architecture supports very extensive Filtering Mechanism that enables Switch Application to set both inclusive and exclusive filters on any field from Layer 2 to Layer 7 of the packet. The SOC Architecture has built in State Machine Driven programmable Rules Engines, also called Fast Filtering Processor, which enables setting inclusive or exclusive filters on any field of any layer (layer 2 to layer 7) of the packet. SOC has the capability to allow all the packets to go through this Filtering Mechanism without sacrificing the line speed switching capability.

Some of the other advanced features supported by the innovative SOC Architecture are

1) Classification of Traffic based on Filtering Mechanism.
2) Policy Based Quality Of Service
3) Load Balancing across Trunk Ports based on Traffic Classification
4) Port Mirroring based on programmable filters thus allowing extensive Mirroring capabilities.

Description of Invention

The SOC architecture comprises seven major components, each of which will be discussed separately. The overall architectural block Diagram is shown in Figure 1 below.

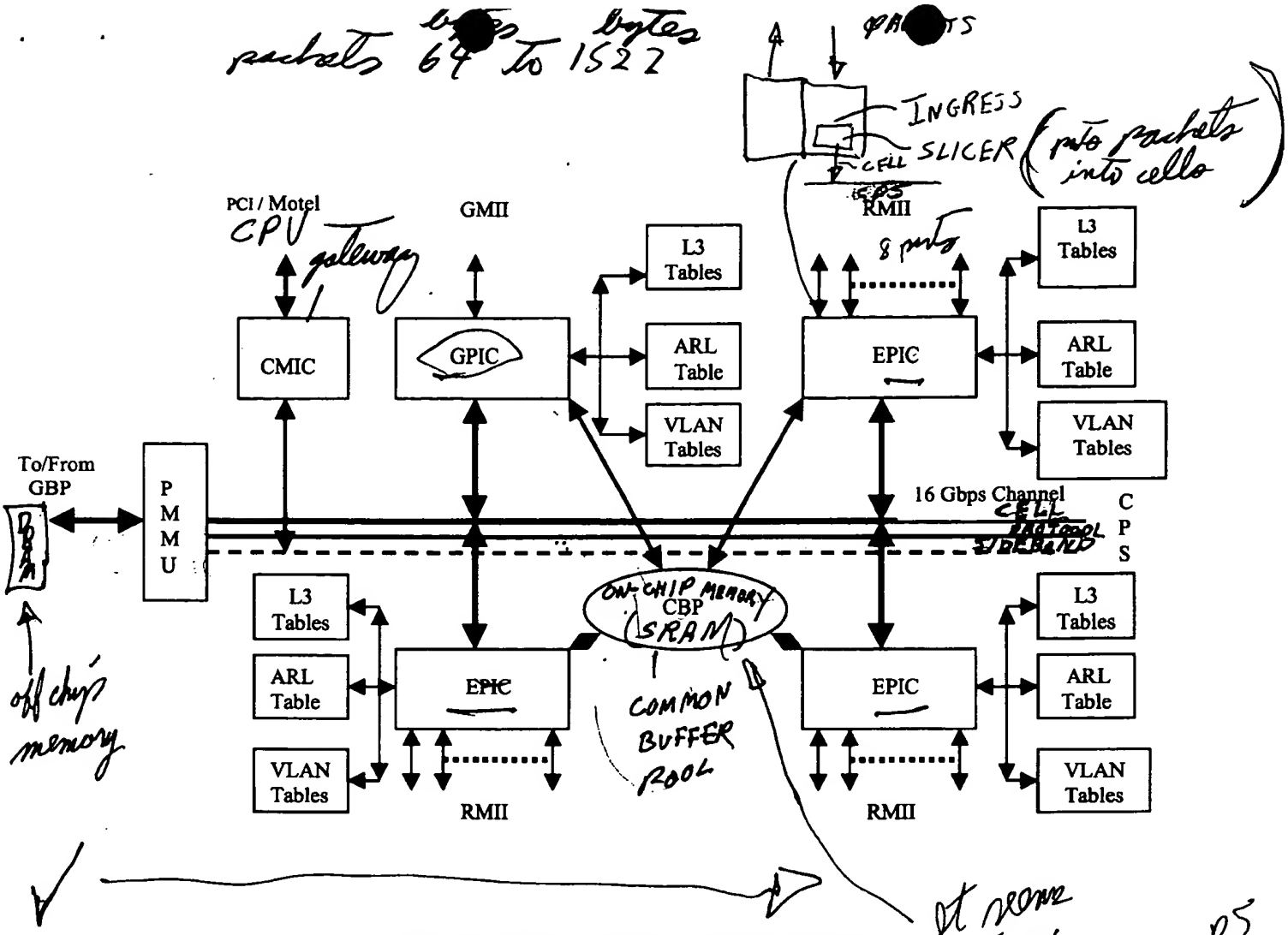


Figure 1: SOC Architectural Block Diagram

The following are the major blocks of SOC:

- Ethernet Port Interface Controller (EPIC)
- Gigabit Port Interface Controller (GPIC)
- CPU Management Interface Controller (CMIC)
- Common Buffer Pool (CBP) / Common Buffer Manager (CBM)
- Global Buffer Pool (GBP)
- Pipelined Memory Management Unit (PMMU)
- Cell Protocol SideBand (CPS) Channel

Ethernet Port Interface Controller (EPIC):

This module interfaces to the 10/100 Ethernet ports. On the medium side it interfaces to the RMII and on the Switch Fabric side it interfaces to the CPS channel. Each EPIC supports 8 10/100 Mbps ports. There are three EPICs' in the SOC. Each EPIC performs both the Ingress and Egress functions. On the Ingress the EPIC supports the following functions:

- L2 Learning (both self and CPU initiated).ARL runs at 132 Mhz.
- L2 Management (Table maintenance including Address Aging)

- L2 Switching (Complete Address Resolution: Unicast, Broadcast/Multicast, Port Mirroring, 802.1Q/802.1P).
- FFP (Fast Filtering Processor including the Rules table)
- Packet Slicer
- Channel Dispatch Unit.

On the Egress the EPIC supports the following functions:

- Packet polling on a per Egress Manager (EgM) / Class Of Service (COS) basis.
- Rerouting / Scheduling
- Head Of Line (HOL) notification
- Packet Aging
- CBM/GBM control
- Cell Reassembly.
- Cell release to FAP (Free Address Pool).
- MAC TX interface.

Gigabit Port Interface Controller (GPIC):

This module is very similar to the EPIC with the following exceptions:

- Supports only one Gigabit Ethernet port.
- ARL Table is not shared with other ports .
- Few other differences like the GMII running at 125 Mhz as compared to RMII running at 50 Mhz.

CPU Management Interface Controller (CMIC):

This block is the gateway to the host CPU. In it's simplest form it provides sequential direct mapped accesses between the CPU and the SOC. The CPU will have accesses to the following resources on chip :

- All MIB counters.
- All programmable registers
- Status and Control registers
- Configuration registers
- ARL tables
- Port Based VLAN Table
- 802.1q VLAN tables
- L3 Tables (Layer-3 IP Tables)
- Port Based VLAN tables
- Rules tables
- CBP Address and Data memory
- GBP Address and Data memory

The bus interface itself will be PCI/PCI64 with Motel as a subset. This way the end user will have the option of using either the PCI or Motel but not both. A "beefed-up" CMIC in addition will include the following:

- Both Master and Target PCI64 (64 bits at 66 Mhz)
- DMA support
- DMA chaining and Scatter-gather.

Common Buffer Pool (CBP) / Common Buffer Manager (CBM):

CBP is the on-chip data memory. This is the first level high speed SRAM. All packets transmitted out of the SOC exit out of this memory. We are shooting for 720 KB on-chip data memory. The CBP runs at 132 MHz. All packets in the CBP are stored as cells.

The CBM does all the queue management. It is responsible for:

- Assigning cell pointers to incoming cells
- Assigning Common Packet IDs (CPID) once the packet is fully written into the CBP
- Management of the on-chip Free Address Pointer pool (FAP).
- Actual data transfers to/from data pool.
- Memory Budget management

Global Memory Pool (GBP):

All re-routed packets end up in the GBP. Re-route signaling is handled by the respective Egress Managers. The GBP is the second level memory and is slower than the CBP. The GBP like the CBP is tightly coupled to the GBM/PMMU. The architecture supports maximum of 64 MB of memory. Like in the CBP the packets are also stored as cells in the GBP. For broadcasts and multicasts only one copy of the packet is stored in the GBP.

Pipelined Memory Management Unit (PMMU):

This unit interfaces to the CPS channel on one side and on the other side interfaces to the off-chip memory (GBP). The PMMU includes multiple write and read buffers for optimal memory utilization. The PMMU supports the following functions:

- Global queue management. This includes assignment of cell pointers for rerouted incoming packets, maintenance of the global Free Address Pointer pool (FAP).
- Innovative Cell Management optimized for time.
- Global memory budget management
- GPID assignment and notification to the Egress Manager.
- Write buffer management so that the RX packets are burst written into the GBP
- Read prefetches based on Egress Manger / Class Of Service (COS) requests.
- Smart memory controller.

Cell/Protocol/Sideband (CPS) Channel:

This is a 16 Gbps channel that "glues" the various modules together as shown in figure 1. The CPS actually consists of 3 channels.

Cell or C Channel: This is 128 bits wide and runs at 132 MHz. All packet transfers between ports occur on this channel. There is no overhead on this channel and is used only for data transfers.

Protocol or P Channel: This is synchronous to the C-channel and is locked to it. During Cell transfers the message header is sent via the P-channel by the Initiator (Ingress/PMMU). The P-channel runs at 132 Mhz and is 64 bits wide.

Sideband or S Channel: This channel also runs at 132 Mhz and is 32 bits wide. The following are its functions:

- CPU management: MAC counters, register accesses, memory accesses etc.
- SOC internal flow control: Link updates, out queue full etc

- SOC inter-module messaging: ARL updates, PID exchanges, Data requests etc.

The CPS data flow diagram for a 64-byte packet is shown in figure 2.

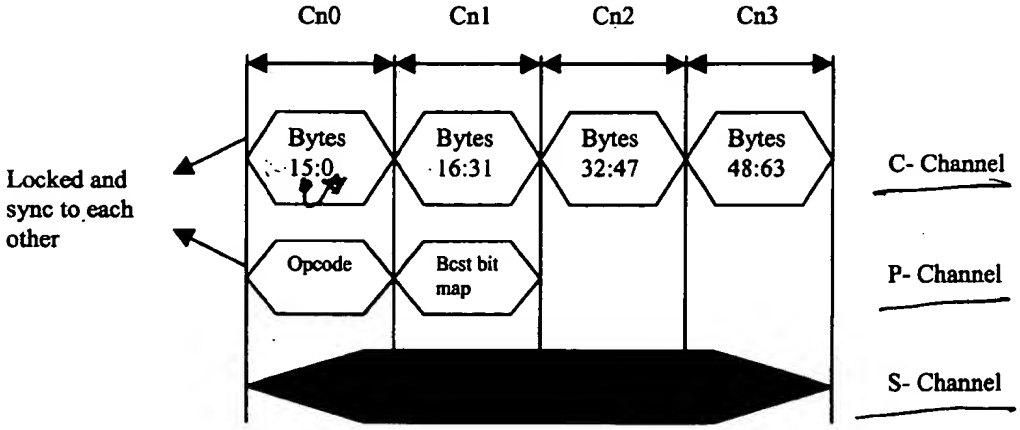


Figure 2: Data flow diagram 64-byte packet.

The Ingress slices the packet into 64-byte cells. Assuming flow-through data writes in the message header into the message buffer based on the ARL result and sets the ready flag in the dispatch unit. The dispatch unit in turn arbitrates for the channel. Upon getting access to the channel writes out the first 16 bytes of the cell into the Channel in phase Cn0 along with the Opcode (unicast or Bcast/Mcast) on the P-channel.. If the opcode is a Multicast or a broadcast the membership bit map is also inserted into the P-channel during phase Cn1 along with bytes 16-31 on the C-channel. During phase Cn2 only data is transferred. During phase Cn3 the PMMU put out the GPID on the P-channel if necessary. During this time and at other times there can be concurrent activity on the S-channel and is decoupled from the activities on the C and P channels. A "start" signal goes active during Cx0 to identify the first transfer. In terms of messaging the first cell of every packet is identified with a S-flag in the message header on the P-channel. The last cell of a packet is identified by the E-flag. If both S and E-flags are active then the packet is 64 bytes in length.

The arbitration on the CPS channel occurs out-of-band. The CPS is a logical bus in the sense that every module can snoop the channel and matching destination ports respond to the transaction. There are 5 masters (excluding CMIC) . In the first phase CMIC will only be a target on the C-channel. The CMIC however can initiate transactions over the S-channel. The C-channel arbitration scheme used is Demand Priority Round-robin. If no requests are active the default TBD module gets to park on the channel. If there is only one requestor that is active then that requestor gets the channel on-demand. If all requests are active then the PMMU gets a grant every other cell cycle. Wherein :

Cell_cycle = 5 * chnl_clock_cycles == 4 data transfers + 1 turn_around.

Chnl_clock_cycle == synchronous to the Master_SOC_clk == 132 Mhz;
The C-Channel arbitration mechanism is shown in Figure 3.

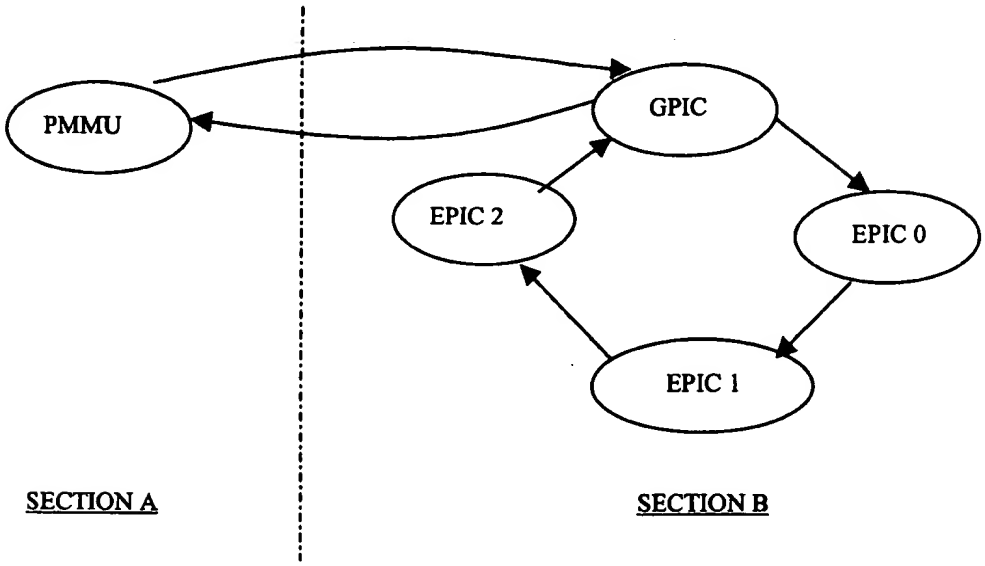


Figure 3: C-Channel Arbitration Mechanism

The C-Channel arbitration is partitioned into two sections. Section A is PMMU and Section B consists GPIC and the three EPIC modules. On a fully loaded channel with all requests active section A and section B get every other cell_cycle. Within section B the accesses to the C-channel are equally shared on a round-robin basis. Example : With all requests active all the time , the C-channel timing is shown in figure 4. This includes turn-around cycles.

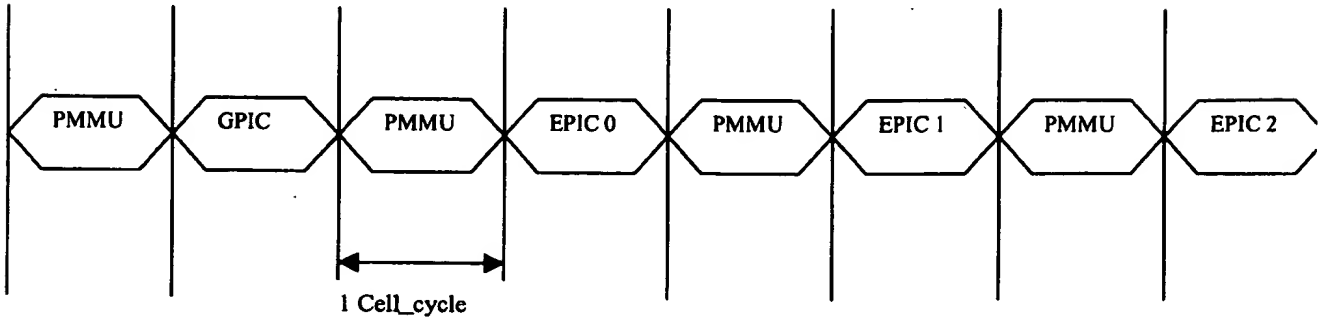


Figure 4: C-Channel Timing

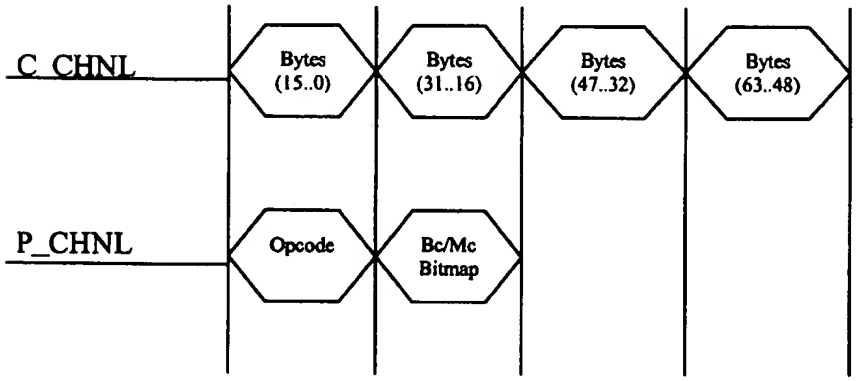
The S-channel is a Sideband channel. As such it is not tied into the C or P channels. The S-channel arbitration is round-robin. The CPU uses the S-channel for accesses to all configuration registers, status/control registers, tables and memory. There are some command requests that go out on the S-channel and the responses come back on the C-channel. Refer to the message section that is described in this specification.

CPS Channel Formats:

Cell Channel Format:

Bytes (15:0)
Bytes (31:16)
Bytes (47:32)
Bytes (63:48)

Cell channel is used for transferring cell data and is always in sync with Protocol Channel.



Protocol Channel Messages

62	60	58	56	54	52	50	48	46	44	42	40	38	36	34	32
Opcode				Dest Port			Src Port			Cos	C	J	S	E	crc	pt	st

30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0
Len			Cell Count				mc	Copy cnt0		Cos	O	Bc Multicast Bitmap (19..26)			

62	60	58	56	54	52	50	48	46	44	42	40	38	36	34	32
Bc Multicast Bitmap (0..18)										UnTagged Bitmap (14..26)					

30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0
UnTagged Bitmap (0..13)							Time Stamp								Res

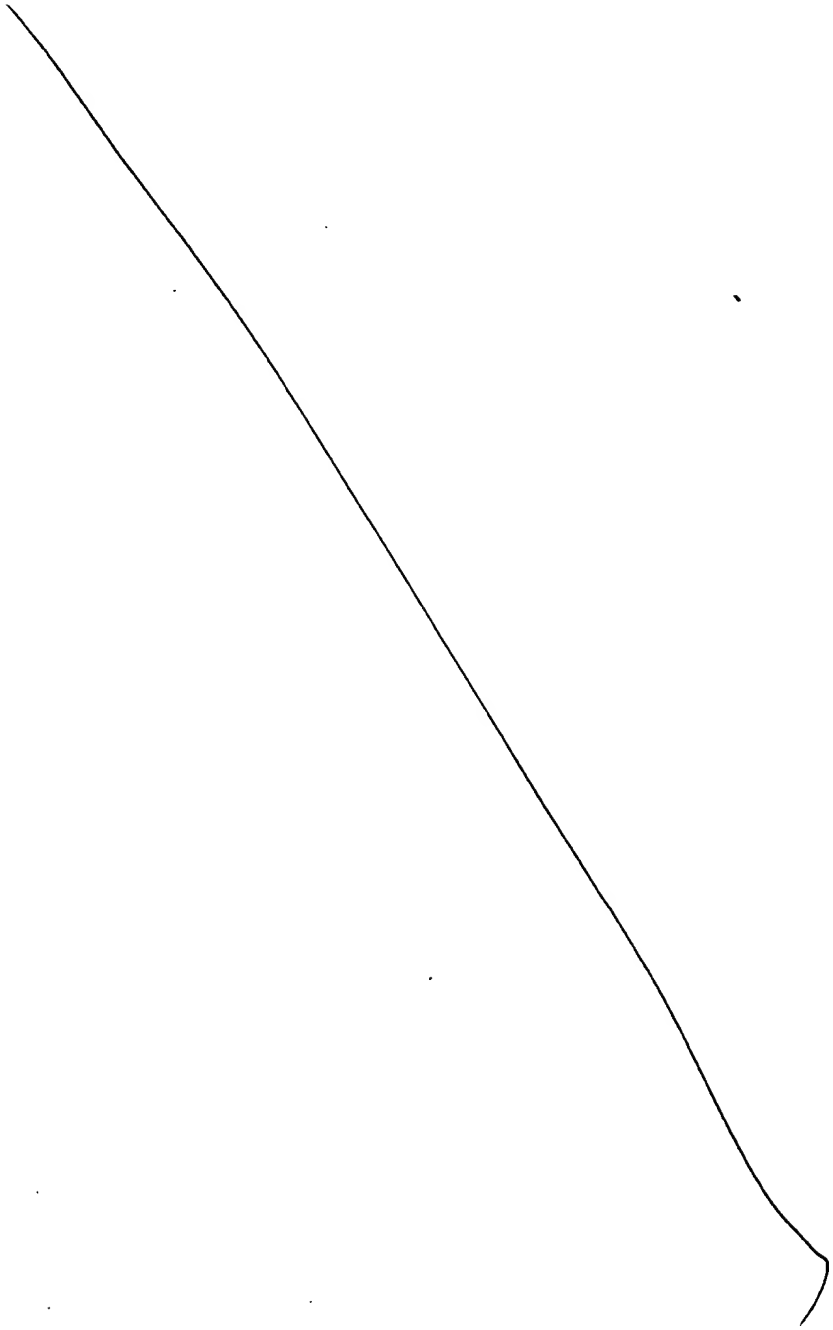
Field Description:

Fields	# of Bits	Description
Opcode	8	Opcode identifies the message
Dest Port	6	Port Number of the destination Port to which this message is addresses to.
Src Port	6	The Port Number which sends the Message
Cos	3	COS – Class of Service for this packet.
C Bit	1	This bit identifies that the destination Port is CPU Port.
J Bit	1	J bit in the message identifies that the Packet is a Jumbo Packet.
S Bit	1	S bit is used to identify that this is the first cell of the Packet.
E Bit	1	E Bit is used to identify that this is the last cell of the Packet.
CRC Bits	2	Bit 0 is Append CRC Bit. If it is set then the egress Port should append the CRC to the packet.

		Bit 1 is Regenerate CRC Bit. If this bit is set then the egress Port should regenerate CRC.
Pt Bits	2	Port Type Bits identifies the type of ingress Port. Value 0 - 10 Mbit Port Value 1 - 100 Mbit Port Value 2 - 1 Gbit Port Value 3 - CPU Port
Status Bit	1	If this bit is set then egress Port should Purge the entire Packet.
Len	6	The Len Bits is used to identify the valid number of bytes in this transfer.
Cell Count	8	Cell Count identifies total number of cells in the Packet.
Mc or Mod Count	2	Mod count is Module Count for this Packet. CPU is also considered as a module. So max value this field can take is 3.
Copy Count 0	5	Total Number of Copies of this Packet for Module 0, that is total number of ports in Module 0, which are suppose to get this Packet.
C Cos	1	C Cos is CPU COS. We support two levels of COS for the CPU. COS 0 is low Priority and COS 1 is High Priority Class Of Service. Cos 1 is used to send control Messages like BPDUs, GMRP, GVRP, etc.
O Bits	2	Optimization Bits are provided for CPU so that it can process the packet more efficiently. Value 1 – is set when the packet is send to the CPU as a result of C Bit set in the Default Router Table.
Bc / Mc Bitmap	27	Broadcast and Multicast Bitmap. This field identifies all the egress ports, the packet should be sent to.
UnTagged Bits	27	UnTagged Bits – This bits identifies all the egress ports which is suppose to Strip the Tag Header.
Time Stamp	16	Time Stamp is a 16 bit running counter, which the system puts in this field when the packet arrives. Time Stamp is implemented with the granularity of 1usec.

The Time Stamp field is valid for the first cell, that is, if the S bit is set. The CRC Bits, Status, Length and cell count fields are valid only for the last cell of the Packet,

when E Bit set to 1.



Protocol Channel Message Types

OpCode Value	Message Types
0x01	Unicast Message
0x02	Broadcast or Multicast Message
0x04	Port Mirroring Message
0x08	Read Data Ack Message
0x10	Early Termination Message

Unicast Message:

This Message is used to transfer Unicast Packets. The Source Port Id identifies the ingress port and Destination Port Id identifies the egress Port.

Broadcast / Multicast Message:

This Message is used to transfer Multicast or Broadcast Packets. Broadcast / Multicast Port Bitmap in the message identifies all the egress ports on which this packet should go out.

Port Mirroring Message:

This message is used to transfer Unicast Packets which has come from a Mirrored Port or which is going to a Mirrored Port. The Port Bitmap in the message identifies the two ports on which the Packet should go out.

Read Data Ack:

This message is used to send the Data from Global Buffer Pool (GBP) to Common Buffer Pool (CBP). The Data itself goes on the Cell Channel. Read Data Ack comes in for a Request sent by CBP on Side Band Channel.

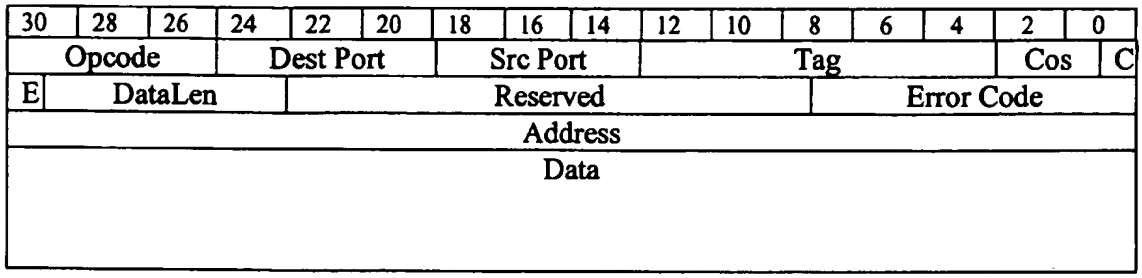
Early Termination:

This message is used to send the Message to the Common Buffer Manager (CBM) to indicate that for some reason the Packet is terminated. In this Message the Status bit should be set to indicate that this is the last cell of the Packet and that the Packet should be purged.

Side Band Channel or S_Channel

The side band channel is 32 bits wide and is used for conveying Port Link Status, Receive Port Full, Port Statistics, ARL Table synchronization, Memory and Register access to CPU and Global Memory Full and Common Memory Full notification.

Side Band Channel Messages



Field Description:

Fields	# of Bits	Description
Opcode	6	Opcode identifies the Message Type
Dest Port	6	Port Number of the destination Port to which this message is addresses to.
Src Port	6	The Port Number which sends this Message
Tag	10	Tag field is an index into Tag array which contains the packet pointer in CBP.
Cos	3	Cos field identifies the Class of Service.
C Bit	1	If this bit is set then this message is for CPU.
Reserved	16	Reserved for future
Error Code	8	Error Code in case of error. Error field is checked only if E bit is set.
DataLen	7	DataLen is total number of data bytes in the message
E Bit	1	E bit is error Bit. It is set if there is an error in executing the Command.
Address	32	Memory Address for reading or writing.
Data	0..127 Bytes	Data Bytes

Side Band Channel Message Types

OpCode Value	Message Types
0x01	Link Up Notification
0x02	Link Down Notification
0x03	COS Queue Full Notification
0x04	COS Queue Available Notification
0x05	CBP Full Notification
0x06	CBP Available Notification
0x07	GBP Full Notification
0x08	GBP Available Notification
0x09	Read Memory Command
0x0a	Read Memory Ack
0x0b	Write Memory Command
0x0c	Write Memory Ack
0x0d	Read Register Command
0x0e	Read Register Ack
0x0f	Write Register Command
0x10	Write Register Ack
0x11	ARL Insert Command
0x12	ARL Insert Complete
0x13	ARL Delete Command
0x14	ARL Delete Complete
0x15	VLAN Insert Command
0x16	VLAN Insert Complete
0x17	VLAN Delete Command
0x18	VLAN Delete Complete
0x19	Rules Table Insert Command
0x1a	Rules Table Insert Complete
0x1b	Rules Table Delete Command
0x1c	Rules Table Delete Complete
0x1d	Get PID Data
0x1e	Release Cell Data
0x1f	Decrement Cell Count
0x20	L3 Insert Command
0x21	L3 Insert Complete
0x22	L3 Delete Command
0x23	L3 Delete Complete
0x24	GPID notification

Advantages of CPS Channel

The advantages offered by this innovative CPS channel are:

1) The SideBand channel is decoupled from C Channel and P Channel, which means that overloading of Cell Channel does not affect the Side Band Channel and vice versa.
2) The Cell Channel and Protocol Channel always runs in sync. The cells are transferred on Cell Channel and Protocol Channel is used to convey the control information of the Packet or the Cell, thus preserving the bandwidth on the C Channel for the Cell transfer.

Functional Operation of ? CBM

When the packet comes in from the ingress port the decision to accept the frame for learning and forwarding is done based on several ingress rules. The ingress does the Address Resolution lookup (ARL) by going through the ARL Tables. In case the packet is addressed to one of the Layer 3 Interfaces of the Switch then it does the L3 and Default Table Lookup. Depending on these lookup the packet the Egress Port is decided.

The Egress block diagram is shown figure 4. The resolved packets are put out on the channel by the Ingress. The CBM interfaces to the channel and every time there is a cell/packet to one of its Egress ports, the CBM gets involved. In terms of the CBP the CBM assigns cell pointers and manages the linked list. The CBM supports several concurrent reassembly engines, one for each Egress Manager, and keeps track of the frame status. Once the packet is fully written into the CBP, the CBM sends out the CPIDs' to the respective Egress Manager. CPID point to the first cell of the packet in the CBP. The Egress Manger control the packet flow to the Transmit MAC once the PID (CPID or GPID) assignment is completed by the CBM. The CBM also decrements the budget register of the respective Egress Manger by the number of cells after the complete packet is written into the CBP.

The Egress Manager writes the PID into its packet pool. If there are multiple Class Of Services (COS) then the Egress Manager writes the PID into the selected COS pool. The Egress Manger has its own scheduler, which interfaces to the Packet FIFO on one side and the packet pool on the other side. The packet pool includes all PIDs'. The Packet FIFO includes only CPIDs'. The packet FIFO interfaces to the TX FIFO and based on the requests from the TX MAC starts off the transmission. Once the transmission starts data is read out from the memory one cell at a time based on the TX FIFO requests.

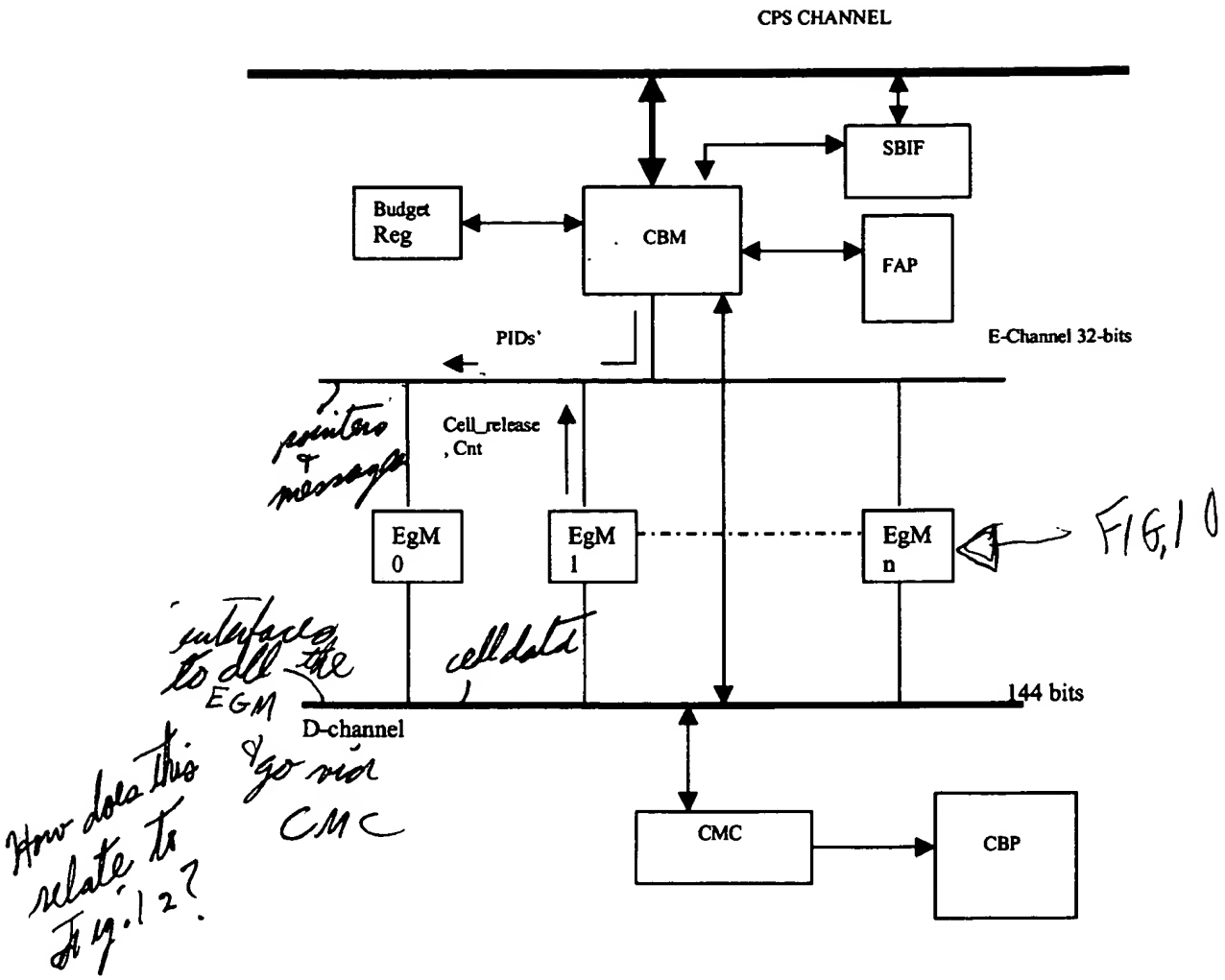


Figure 4: Egress Block Diagram In Detail

Where:

CBP - Common Buffer Pool (on-chip memory)
CBM - Common Buffer Manager
CMC - Common Memory Controller
D_channel - Used for cell data transfers.
E_channel - Used for transfer of pointers and messages between the Egress Manager and CBM.
EgM = Egress Manager (one Egress Manager per port)
FAP = Free Address Pool. Contains CBP free cell pointers.
SBIF = SideBand Interface module. Interfaces to the S-Channel. All messages to/from the Egress go via this module.

For unicast traffic as the cells are read out the Egress Manager sends out cell release pointers to the CBM. After the last cell of the packet is flushed out the EgM sends out a cell release along with a Cell Count (cell_count) value. The CBM uses this value to decrement the budget register. For broadcast/multicast traffic the cells are released by the last member Egress Manager reading out the packet.

The CMC (CBP Memory Controller) interfaces to both the CBM and the EgMs. The memory access arbiter is part of the CMC. The CBM generally uses the CMC for doing writes into the CBP. The EgM uses the CMC for reading out data from the CBP. There is one exception here wherein if the traffic is multicast or broadcast the EgM executes a read-modify write on the copy_cnt field.

Common Buffer Manger:

As stated earlier CBM performs the following functions:

1) On-Chip FAP Management
2) Memory Budget Management
3) Channel Interface
4) Cell Pointer Assignment on a per Egress Manager /Class of service basis.

These functions are described in greater detail below.

On-chip FAP management:

The CBM manages the FAP and as such assign free cell pointers to the incoming cells and writes back to the FAP the released cell pointers from the various Egress Managers. Assuming there is enough CBP space available and enough Free Address Pointers available, the CBM in its local buffer keeps at least 2 cell pointers per Egress Manager per Class Of Service. When the first cell of a packet arrives to an Egress Manager, the CBM writes this cell to the CBM memory location at the address pointed to by the first pointer. In the next cell header (Next_Cell_Header) field it writes in the second pointer. The format of the cell as it is stored in the CBP is shown in figure 5. Each line is 18 bytes wide.

Line 0	FC | LC | BC/MC | Cpy_cnt(5b) | Cell_length (6b) | CRC (2b) | NC_header (16b) | Cell_cnt(8b) | Time_Stamp (2B) | reserved (1B) Cell_data (0-9B)
Line 1	Cell_data (10-27) Bytes
Line 2	Cell_data (28-45) Bytes
Line 3	Cell_data (46-63) Bytes

Figure 5: CBP Cell Format

Where:

FC - First Cell, marks the first cell of a packet
LC - Last Cell, if both FC and LC are set then this is a single cell packet. Minimum packet size (64 Bytes).
BC/MC - Broadcast/multicast flag.
Cpy_cnt - Valid only if BC/MC flag is set. Identifies the number of Egress ports that are part of the Broadcast or Mutlicast.
Cell_length - Number of valid cell bytes in this cell. "0" means only byte 0 is valid, "1" if byte 1 is valid, and so on.
Type = 00, reserved
= 01, Append CRC
= 10, Regenerate CRC
= 11, reserved
NC_header - Next_Cell header, pointer to the next cell. Valid only if LC = 0.
Cell_cnt - Valid only if LC = 1. Total cell count in the packet. Used for memory budgeting.
Time_Stamp - 2 bytes, runs of a programmable clock which is a multiple of the system clock.
Cell_data - Stored as 64-byte cells

The FAP stores all the free pointers for the CBP. The depth of the FAP is 8k-12k pointers, in terms of bytes this translates to 16 Kbytes to 24 Kbytes. Each pointer in the FAP points to a 64-byte cell in the CBP. The actual cell stored in the CBP is 72 bytes, 64 (byte data) + 8-byte control info.

Memory Budget Management:
There are three cases :

- Egress out queue budget
- Egress Packet Pool updates
- Ingress (RX port) budget.

The Egress out queue budget is used for HOL (Head Of Line blocking). This budget register is 16 bits wide and the value represents the actual out queue length (in cells) at any instant of time. There is also a HOL high water mark register per Egress that activates the port disable on the Ingress once that particular Egress hits the high water mark. Refer to the messages section for the actual message and its format. There is also a similar HOL low water mark register per Egress that enables a disabled port once the out queue length goes below the low water mark. The out_queue_budget register default value is 0. Every time the CBM assembles a packet in the CBP and sends a CPID assignment to a Egress, that Egress out queue budget register gets incremented by the cell count. Similarly every time a EgM sends out a packet it notifies the CBM with a cell_release, cell_cnt message. The CBM then decrements that particular Egress out_queue_budget register by the cell _cnt. The format of the above registers are shown in figure 6.

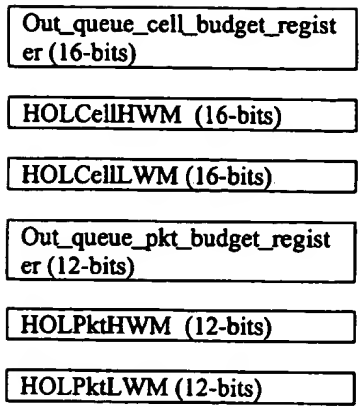


Figure 6. HOL Management registers

Egress Packet Pool budgeting is used to monitor the depth of the Egress Manager packet pool. Each Egress Manager has its own packet pointer pool that stores pointers to the first cell of the packet. The packets (data) can be sitting in either the CBP or GBP. Every Egress Manager is assigned a Packet Pointer Budget Register (pp_budget_register). Every time a packet pointer is written into the packet pointer pool, the pp_budget_register gets incremented, and every time the Egress Manager reads out the pointer the pp_budget_register gets decremented by 1. The Egress Manager manages this register. In terms of flow-control, any time the pp_discard limit is reached, the CBM via the SBIF sends out a COS Queue Notification message via the S-channel to all the Ingress ports. Similarly once the packet pool pointer depth goes below the pp_lwm , a COS Queue Available Notification message is sent out via the S-channel to all the Ingress ports.

Rx_budget updates are used for flow-control by applying back pressure via the Ingress ports. The CBM whenever it successfully transmits a packet out of CBP sends out Decrement Cell Count message via the SBIF. The designated ingress picks up this message and decrements its Rx_budget_reg by the cell_cnt that is contained in the message. Each Ingress Port increments its own Rx_budget_reg by the cell_cnt after successful reception of the packet.

Channel interface:

The CBM constantly monitors the CP channel and any time there is a cell transfer to one of it's Egress Manager, it picks up the cell. It reads in the cell only if there is enough CBP memory space is available. The CBM is always a target on the CP channel. It initiates transactions like get_pid_data on the S-channel via the SBIF. For unicast traffic, the EgM is identified by the destination_port_id that is part of the cell transfer message on the CP channel. For Broadcasts/Multicasts traffic, the EgMs are

identified by the port_group_membership. Refer to the messages section for the format of these messages.

Cell pointer assignment on a per Egress Manager / Class Of Service basis:

For every Ingress, the CBM maintains two buffers each containing a CBP cell pointer. If there are 25 Ingress ports each supporting 8 Classes Of service. So there is 200 of these 2-deep buffers. As long as the quota for a particular EgM/ COS exists, the buffers are filled from the FAP. The flowchart for the cell admission and management is shown in figure 7. For every new packet a reassembly engine is started by the CBM. The reassembly engine gets closed only after the complete packet is written into the memory. In the case of packets admitted into the CBP, the CBM assigns the CPID to the packet and sends a message to the respective Egress Manager. The following registers are used in the CBM :

General CBM registers :

- Maximum_CPB_Space (MCS), Static-programmable, value in cells
- Curent_CBP_Space (CCS), Dynamic, value in cells.

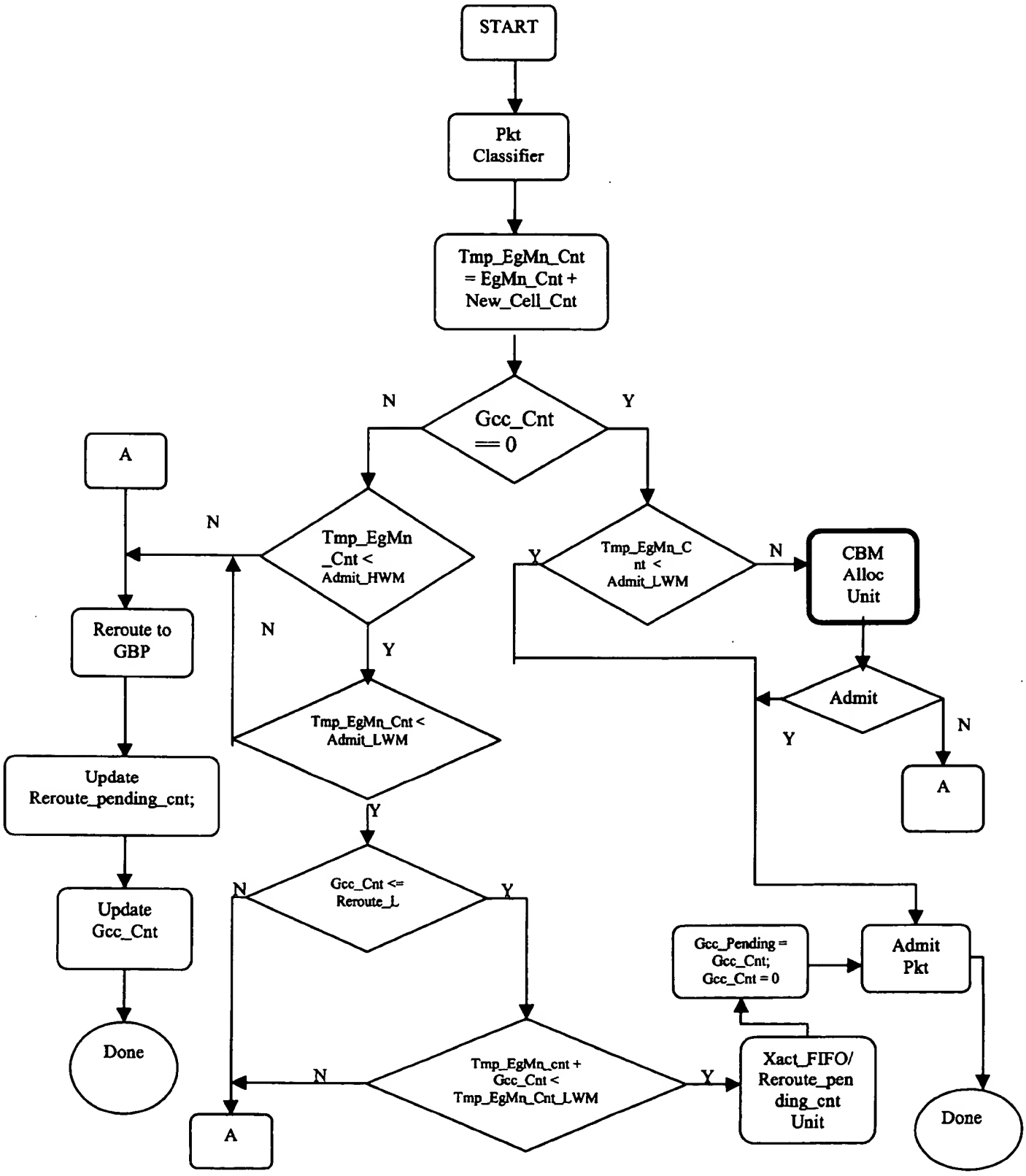
Per Ingress :

- Cell_Count_Register (CCR). The current cell count of the packet being assembled in the reassembly engine. After the packet is fully assembled in the CBP this value gets inserted along with the last cell write.
- NP = Normal packet (up to 1518 bytes deep). This is a 1-bit flag used by the CBP for packet management also gets inserted into the Egress Manager packet pointer pool once the packet gets assembled.
- JP = Jumbo packet (up to 9020 bytes) . This is also a 1-bit flag.
- D = Drop packet flag. 1-bit

Per Egress / COS :

- CBP_Memory_Alloc_minimum (CMAmin). This is a programmable register, value in cells.
- CBP_Memory_Alloc_maximum (CMAmax). Maximum CBP memory allowed for this Egress, in cells.
- Current_Memory_Count (CMC). Dynamically updated, run-time CBP memory count for this Egress in cells.
- GBP_Cell_Count (GCC). The number of cells currently in the GBP for this Egress.

Figure 7: CBP ADMISSION LOGIC IN DETAIL



GCC_Cnt = Current count of cells in GBP.
Admit_LWM = Enables reception of new packets into the CBP if the total number of cells in the Egn (Egress n) is below this cell count. This being true by itself is not sufficient enough to allow the packet into the CBP.
Admit_HWM = Disable reception of new packets above this count in the CBP.
GCC_Pending_Cnt = Temp register to hold GCC_Cnt. Used during the re-admission process of new packets (from Ingress) directly into the CBP.
Tmp_EgMn_Cnt = Egress Manager n , current cell count.
Reroute_L = Programmable register . Enables admission of new packets into the CBP only if GCC_Cnt < Reroute_L. This being true by itself is not sufficient enough to allow the packet into the CBP.
Reroute_pending_cnt = Number of packets rerouted and still waiting for GPID assignment.

The Egress Manager Scheduler is part of the Egress manager and is discussed later. The Reclaim unit is used in cases where the packet gets dropped due to the P (Purge) bit getting set. In this case the Reclaim unit cleans up the memory by flushing out the dirty cells of the packet and writing back the cell pointers into the FAP. In order for this to occur the first cell pointer needs to be stored till the whole packet gets written into the memory. The Reclaim unit is illustrated in Figure 8.

the unit isn't shown — its functional operation is shown

Should you show something that shows the relationship of the Egress Manager & its component parts
- Scheduler
- Reclaim unit
- anything else ?

Figs. 10 & 11 shows scheduler but what about reclaim unit ?

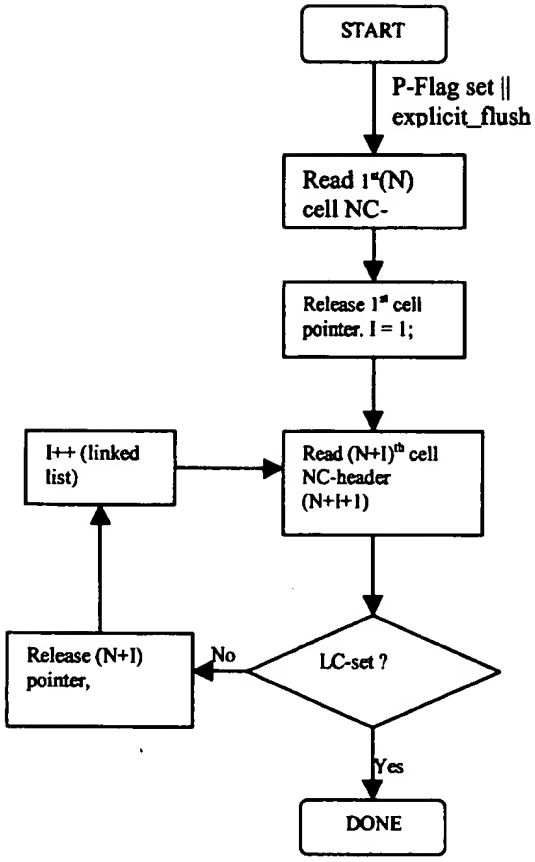


Figure 8: CBM Reclaim Unit

The CBM locally stores the first cell pointer (N) in its register until the packet gets fully assembled in the memory. If the P-flag gets set during this time, then the CBM sends a command along with cell pointer N to the Reclaim unit. The Reclaim unit then scrubs the linked list starting with cell pointer N and starts releasing the pointers to the FAP until the last cell (inclusive). Also, when the P-flag gets set the CBM does not write the PID into the Egress Manager packet pointer pool. The Reclaim unit is also used by the EgM to flush out aged out and heavily congested packets from the out queues. In these cases, the Egress Mangers send out explicit_flush messages along with the first cell pointers. Just like in the P-flag case, the cell pointers get released to the FAP and the memory budget is also adjusted to reflect the changes.

The CBP write unit (CWU) example is shown in Figure 9. This unit keeps track of every frame going to the CBP on a per Ingress basis. For every new frame it keeps the first cell pointer (N) and the next cell pointer (N+1). The CWU maintains the linked list

during the entire frame active time. The actual data write into the CBP is executed by the CMC based on the cell pointers supplied by the CWU.

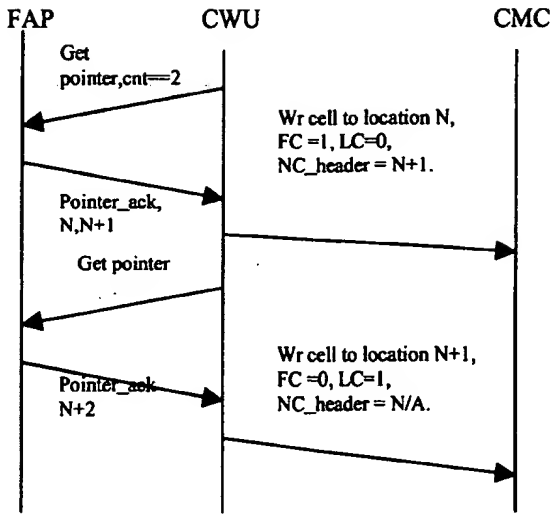


Figure 9: CBP Write Unit (CWU) data flow, 128 byte transfer

Egress Manager :

Once a PID is entered into the Egress packet pointer pool, it is the responsibility of the Egress Manager to handle the data flow until the packet gets read out by the MAC. The block diagram of the Egress Manager is shown in Figure 10. There is one EgM per port. The EgM receives pointer information from the CBM. For unicast packets (no port mirroring) there is only one EgM recipient and for multicast/broadcast there can be several recipients of the pointer information. However in the case of multicast/broadcast all member ports get assigned the same PID in order to avoid multiple copies of the same packet. The EgM can be broken down into 3 stages.

The first stage includes the ECIF (E-Channel Interface) and the **Transaction FIFO**. The ECIF handles the message passing between the EgM and the CBM. The Transaction FIFO is a fixed depth FIFO and stores pointers for both unicast and multicast/broadcast. This interfaces to the E-channel and picks up the pointer messages assigned directed to this Egress. The pointer messages are stored in the following format :

Transaction FIFO entry :

G/L (1b) | JP (1b) | NP (1b) | PID [19:0] = G ? GPID :

The second stage includes the **EgM Scheduler, COS Manager** and the **Packet FIFO**. The Packet FIFO stores CBP Packet pointers (CPID). This FIFO requests CPIDs from the scheduler. The **Scheduler** consults the **COS Manager** and depending on the decision of COS Manager decides on the Priority Queue from which to pick up the next Packet Pointer. The Scheduler then reads out the PID from the selected Priority Queue of the Transaction FIFO (TransF) and acks the Pkt_FIFO once the packet gets assembled in the CBP. Every valid entry in the Pkt_FIFO is a CPID. A top level state diagram of the Scheduler is shown in figure 11.

The third stage is the TX_out stage. This includes the Memory Read Unit (MRU), Timestamp Check Unit (TCU) , MAC_FIFO and the Accelerated Packet Flush (APF) unit. The MRU reads out cells from the CBP (via CMC) based on requests from the MAC_FIFO. The MRU after reading the first cell of the packet passes on the Timestamp field (stored along with the cell in the CBP) to the TCU unit. Only after passing the TCU check the packet gets transferred to the MAC_FIFO.

The TCU manages packet aging. The TCU contains the following registers :

Current Time Register (CTR):

16-bit timer, . Runs off the same clock as the

Discard Packet Register (DCR):

16-bit programmable .

Rule : If (CTR – TimeStamp) >= DCR
Then discard_packet and increment Pkt_discard_age register ;

Pkt_discard_age_register :

32 bit register. Default = 0. Increments every time a packet gets discarded due to aging. Used by

The APF monitors the Pkt_FIFO and any time the FIFO hits full starts off a programmable built-in timer. Upon expiration of the timer flushes out the Pkt_FIFO. The APF interfaces to the Reclaim Unit in the CBM. The APF sends out a disable_port message to the Ingress ports once the built_in timer expires. The APF timer register is shown below :

16-bit APF timer . runs off the same clock as the CTR.

The Untag unit sniffs the first cell of every packet being read into the MRU. If the U-flag is set in the cell then the Untag unit removes the 802.1q tag header before the packet gets dispatched to the MAC_FIFO. After removal of the tag if the resulting packet size turns out to be less than 64 bytes, then extra bytes need to be padded to make the resulting packet size 64 bytes. After tag removal the Untag unit should signal the MAC to recalculate the FCS. Both padding and recalculation of the FCS should be performed by the MAC.

The MAC_FIFO is a shallow FIFO and interfaces to the TX_MAC on the medium side. This FIFO has programmable thresholds for request data. No pointers are passed between this FIFO and the MRU. The MRU keeps track of the linked list and prefetches the data. The MRU flags the beginning and the end of the packet to the MAC_FIFO. In cases wherein there are excessive collisions (16 retries) it is the responsibility of the MAC to read out the entire packet and discard it. The MAC in these cases will also update its excessive collisions register. The MAC should similarly flush out packets with excessive deferrals. These are packets waiting for transmission longer than two max packet times.

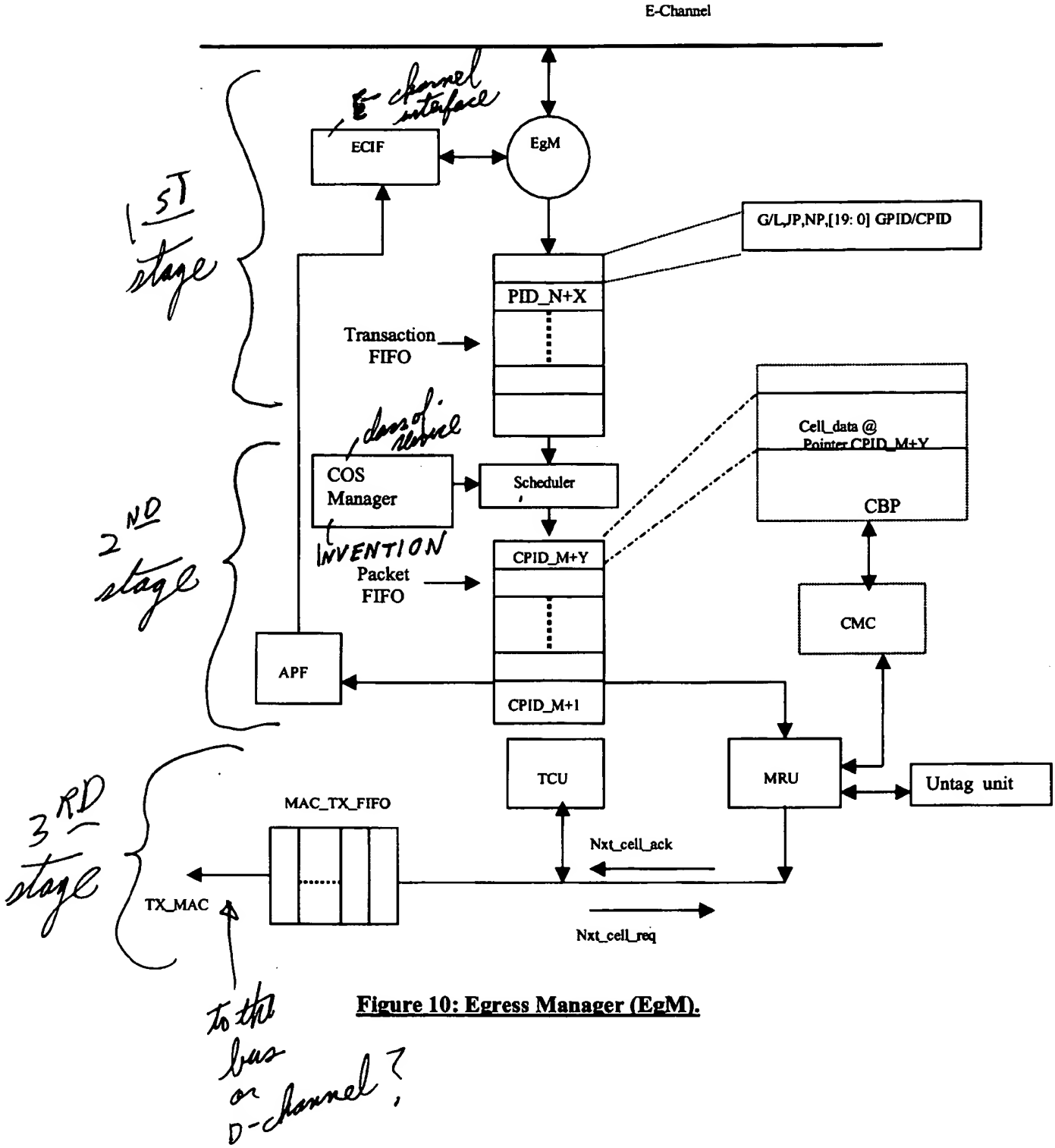


Figure 10: Egress Manager (EgM).

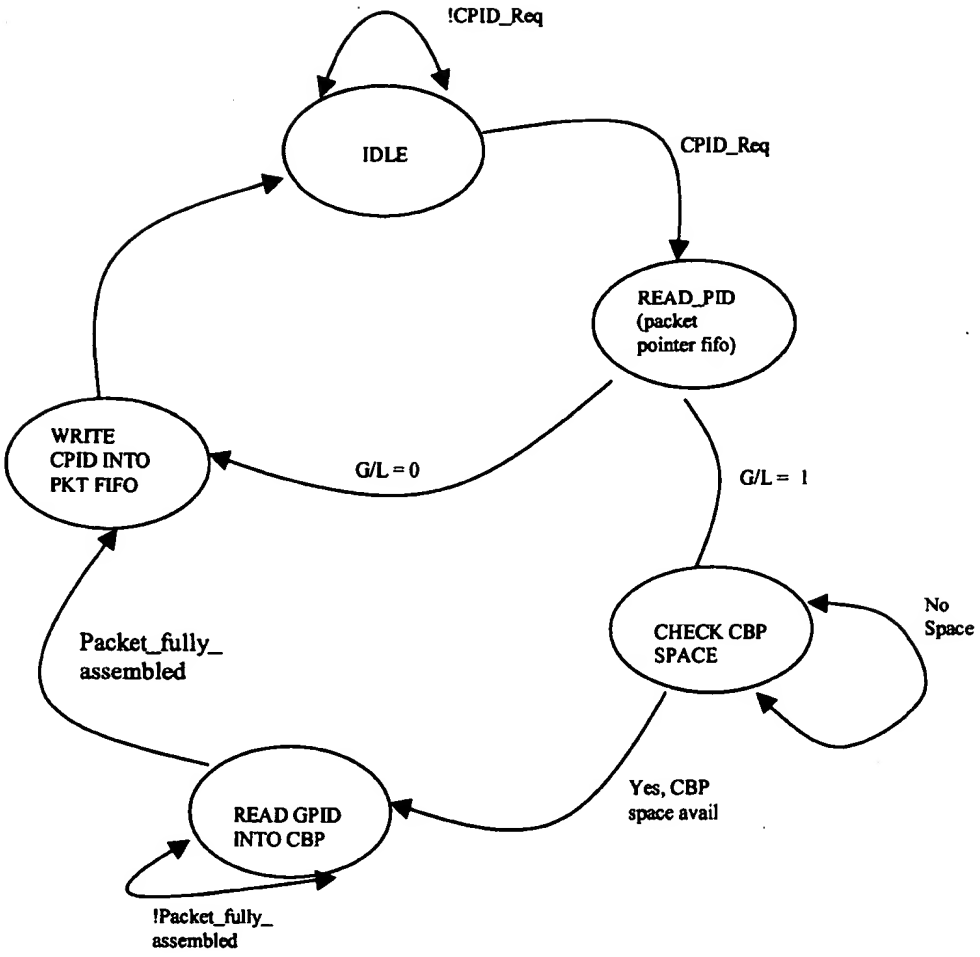


Figure 11: Top level EgM Scheduler State diagram

CBP Memory Controller (CMC):

All modules and the CBM interface to the CMC. The CMC has separate read and write channels to the on-chip SRAM. The CMC interface block diagram is shown in figure 12.

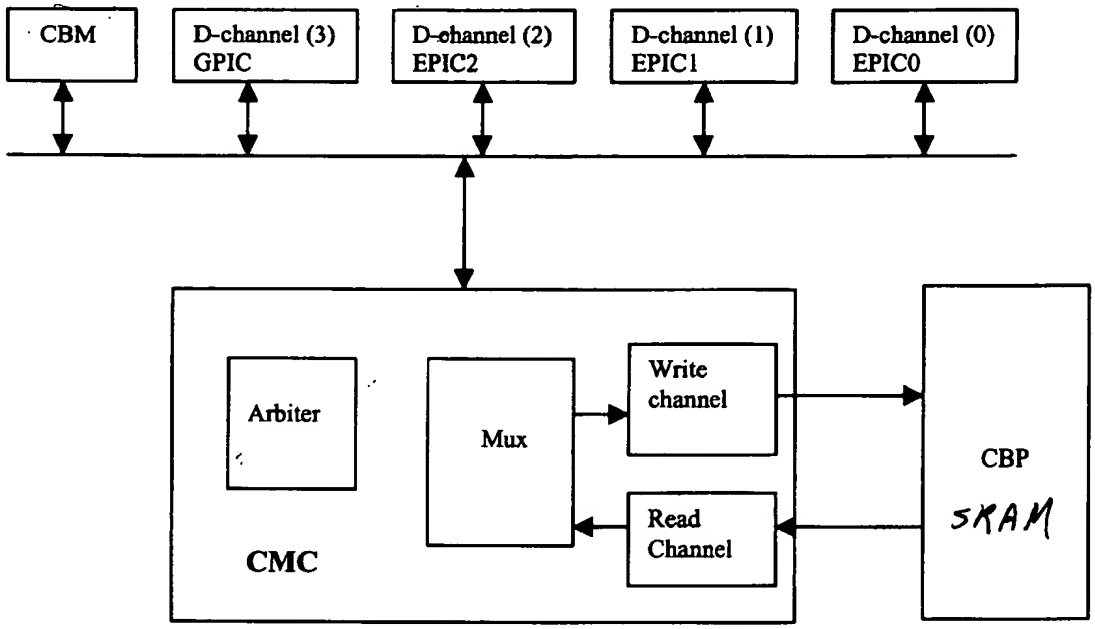


Figure 12: CMC Interface

The top-level Arbiter state diagram is shown in Figure 13.

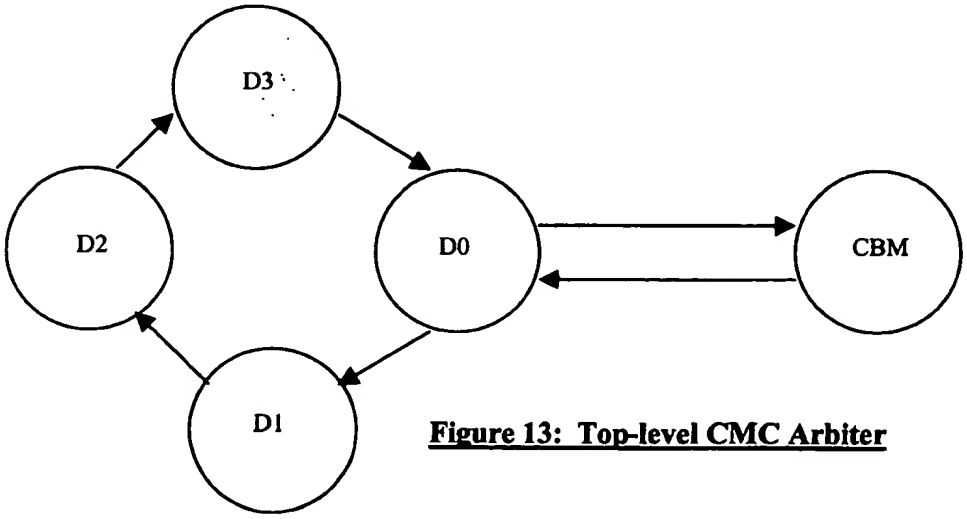


Figure 13: Top-level CMC Arbiter

COS Manager:

The COS Manager is another innovative Logic Module in the SOC Architecture which needs some explanation. COS Manager enables the capability of Policy Based Quality Of Service.

The packets (packet pointers really) depending on the type of traffic (this decision is really made at the Ingress) lands up in Transaction FIFO Priority Queues. The Scheduler depending on the decision of the COS Manager decides to pick up the next packet from one of the Priority Queues. COS Manager can be programmed to enable different types of Queue Scheduling Algorithms.

Strictly Priority Based Scheduling:

If there are any packets residing in the High Priority Queue of the Transaction FIFO, then they are taken up first for transmission. The main disadvantage of this scheme is starvation of low priority queues.

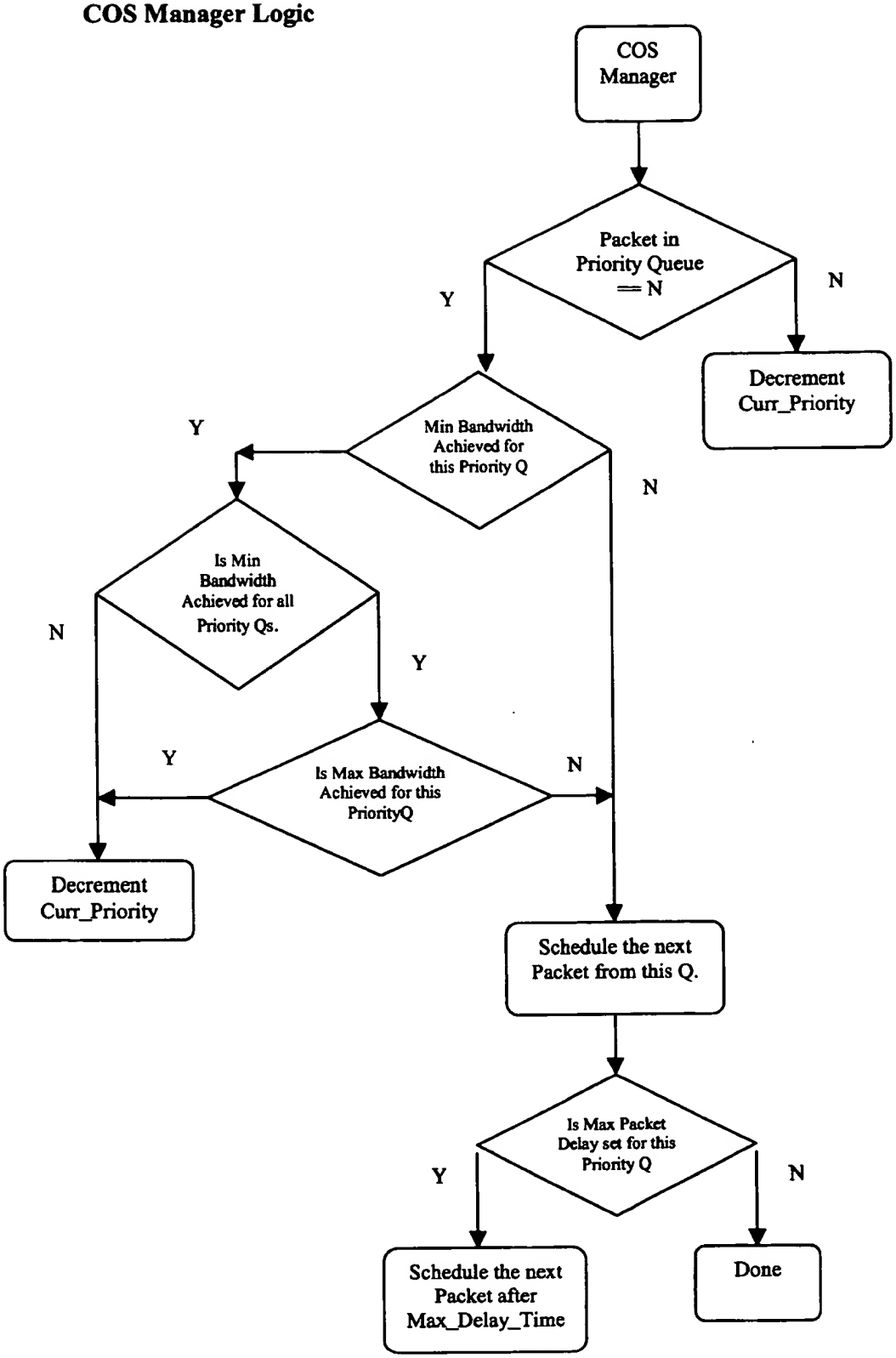
Weighted Priority Based Scheduling:

This scheme alleviates the disadvantage of the Strictly Priority Based Scheduling Scheme by providing Minimum Bandwidth to all the Queues, so that none of the Queues gets starved. The Minimum Bandwidth is really a programmable register in the COS Manager and is programmed by the Switch Application. After achieving the requirement of the Minimum Bandwidth allocation on all the Queues, the COS Manager for the remaining bandwidth checks if any of the Priority Queues has exceeded the Maximum Allocated Bandwidth of the Queue. This ability gives more control to the network Manager to control the Bandwidth per application. The COS Manager also accepts the third parameter per Priority Queue and that is - the Maximum Packet Delay. The COS Manager uses this parameter for scheduling the packet transmission such that the packet on this queue are not delayed more than the Maximum Packet Delay Time. This parameter is mainly useful for real time traffic like Audio and Video.

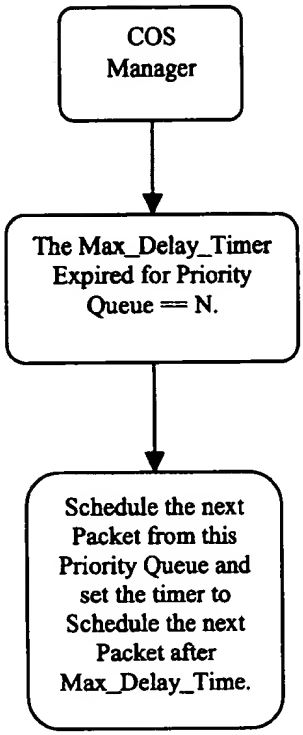
The Programmable Registers associated with each PriorityQueue are

1) Priority Queue Control Register - is used to select the Priority Scheme per Egress Port.
2) Minimum Bandwidth Register – These are 8 Register per Egress Port and the Bandwidth is expressed as percentage of total bandwidth.
3) Maximum Bandwidth Register – These are 8 Register per Egress Port and the Bandwidth is expressed as percentage of Total bandwidth.
4) Maximum Packet Delay – is expressed in microseconds. Again these are 8 registers per Egress Port.

COS Manager Logic



COS Manger Logic When Max Delay Timer Expires for a Priority Queue



Fast Filtering Processor

SOC Architecture supports very extensive Filtering Mechanism that enables Switch Application to set both inclusive and exclusive filters on any field from Layer 2 to Layer 7 of the packet. The SOC Architecture has built in State Machine Driven programmable Rules Engines, also called Fast Filtering Processor, which enables setting inclusive or exclusive filters on any field of any layer (layer 2 to layer 7) of the packet.

The filter itself is 64 bytes wide and can be applied on an incoming packet starting from any offset. This gives flexibility for applying filter on any protocol field. Various actions are defined in the rules database. The actions may involve 1) 802.1p Tag Insertion, 2) 802.1p Priority Mapping, 3) IP Type Of Service (TOS) Tag Insertion, 4) Event to CPU 5) Discard the packet and 6) decide the egress Port (This feature is used for Load Balancing), 7) send the packet to the Mirrored Port. The Combinations of all the above actions is also supported.

Filter Database

Pointer Offset (14 Bits)

Egress Port Mask (5b)	Ingress Port Mask (5b)	Inclusive Filter Mask (512 Bits)

Egress Port Mask (5b)	Ingress Port Mask (5b)	Exclusive Filter Mask (512 Bits)

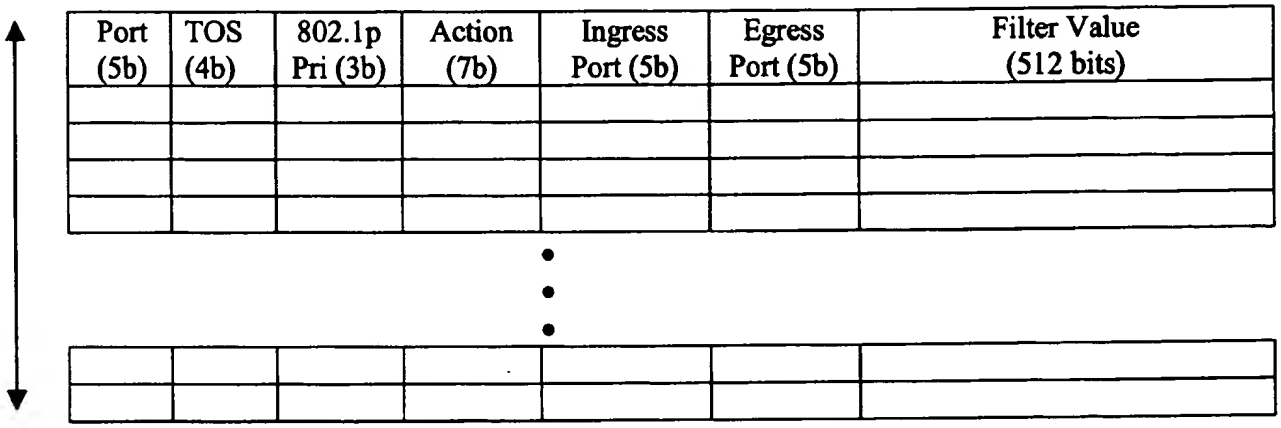
The filter database has 8 sets of Pointer Offset, Inclusive Filter Mask and Exclusive Filter Mask Registers. Once the Inclusive Filter Mask and Exclusive Filter Mask is applied to an incoming packet at the given offset the result is compared with the entries in the Inclusive Rules Table and Exclusive Rules Tables respectively. If there is a match on an entry in the Inclusive Rules Table, then actions are picked from that entry and executed on the packet. If there is no match in the Exclusive Rules Table then the packet is discarded, otherwise the actions are picked up from the matched entry and executed on the packet. Ingress Port Mask or egress Port Mask is set only if, one intends do filtering on a per port basis. In that case the ingress port or egress port is used along with the data result to do the comparison in the Rules Tables.

Rules Table

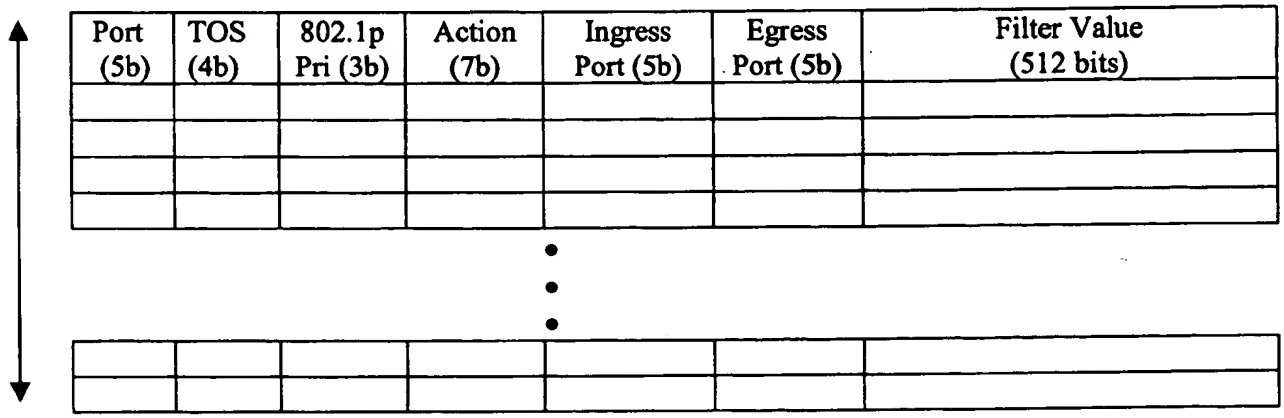
The Rules table itself is 128 entries deep, but is partitioned for inclusive Filters and Exclusive Filters. Out of 128 entries, first 96 entries are used for inclusive filters and remaining 32 entries are for exclusive filters. The entries in both the rules tables, inclusive and exclusive, are stored in ascending order with Data Result + Egress Port + Ingress Port as the key. The Ingress Port or Egress Port is set only if there is intention to do the filtering per port basis and in that case the Ingress or Egress Port Mask should be set to 0xFF.

Rules Table Formats

Inclusive Table Format



Exclusive Table Format



Rules Table Fields

Fields	# of Bits	Description
Filter Value	512	Filter value
Ingress Port	5	Ingress Port Number : This field is set only if one is setting this filter on a specific ingress port. If this field is set then the Egress Port Mask in the Filter Register should be set.
Egress Port	5	Egress Port Number : This field is set only if one is setting this filter on a specific egress port. If this field is set then the Egress Port Mask in the Filter Register should be set.
Action Bits	7	Action Bits defines the actions to be taken in case of the matched entry. Bit 0 – If this bit is set then insert 802.1p Priority Tag in the packet. The Priority is picked up from the 802.1p priority field. Bit 1 – If this bit is set then categorize this packet to send on priority COS, but don't modify the packet with 802.1p priority tagged header. Again the priority is picked up from the 802.1p Priority field. Bit 2 – If this bit is set then change IP TOS in the IP Header. The new TOS value is picked up from the TOS field. Bit 3 – if this bit is set then send the packet to CPU. Bit 4 – if this bit is set then discard the packet. Bit 5 – If this bit is set then select the output port from the Port Field. Bit 6 – If this bit is set then the packet is sent to the mirrored port.
802.1p Priority Bits	3	The value in this field is used to assign the priority to the packet. The 802.1p standard defines 8 levels of priorities from 0 to 7. The field is used only if bit 0 or bit 1 of Action Field is set.
TOS field	4	The value in this field is used to assign the new value to TOS field in the IP Header. This field is used only if bit 2 is set.
Output Port	5	This field identifies the output Port Number. This port overides the egress port selected by ARL. It is advisable to use this feature along with Trunking. It is also the responsibility of software to set this port to be one of the Trunk

		Ports

The Fast Filtering Processor is used to support the following feature

1) Classification Of Traffic
2) Load Balancing across Trunk Ports based on Traffic Classification
3) Port Mirroring based on programmable filters

Classification Of Traffic:

The Filtering Mechanism enables the SOC to classify traffic in variety of way. The Filtering Processor can modify the packet so as to add the Tag header. The Tag Header contains the priority field, which should be set to the value decided by the Filtering Rules. The Ingress sends the packet to the Egress Manager with the COS so that the packet goes to the Priority Queue decided by the Filter Rules.

Load Balancing across Trunk Ports based on Traffic Classification:

The Filtering Mechanism also provides the feature to do the load balancing depending on the traffic classification. The Filter rules are set such that the match on the certain protocol fields in the packet enables the Filter processor to select ~~certain~~ Egress Manager.

predetermined

Port Mirroring based on programmable filters:

The Filtering Mechanism also allows the SOC to send the packet to the Mirrored port depending on the filter. The Filter rules can be set such that the packet is forwarded to Mirrored Port only if frames it comes from certain ingress port and is going out on certain egress port.

Doesn't read well

Innovative Layer 3 Switching Implementation.

SOC supports Layer 3 Switching only for IP Protocol under certain conditions. In case of Layer 3 Switching, CPU plays an important role. Even though SOC offloads CPU in Layer 3 Switching for IP Protocol, CPU is still involved in the following functions.

1) Running RIP, RIP2, OSPF or any other Routing Protocol to generate the Routing Tables.
2) Running ARP Protocol to resolve the IP Address and to generate and maintain ARP Table.
3) Setting up the L3 table, which will be used by SOC for Layer3 Switching.

L3 Switching Configuration Details

L3 Switching is enabled by configuring specific L3 interfaces. L3 interfaces are configured with the following information:

1) L3 interface identifier (index)
2) IP Address
3) Subnet Mask (if appropriate)
4) Broadcast Address
4) MAC Address
5) VLAN ID

L3 interfaces (using their unique MAC addresses) can be addressed by end systems to send packets off the local Subnet. Multiple L3 interfaces can be configured per Virtual LAN (VLAN), but there can be only one L3 interface per IP subnet.

L3 interfaces are not inherently associated with a physical port, but with VLANs. If a VLAN is defined to be limited to a single physical port, then effectively the classical router model of L3 interfaces per physical port can be imitated.

Up to 32 L3 interfaces can be configured per SOC.

The L3 Switching, the way it is provided by SOC, optimizes the implementation for *delivery* of packets between subnets in VLANs physically connected to the switch, and (optionally) *forwarding* of all other packets to a pre-designated or CPU-controlled default router. If the forwarding option is not chosen, all *forwarding* of packets to remote subnets is performed by software running on the associated CPU.

L3 Switching in Detail

When packets arrive destined to a MAC address which is associated with an L3 interface for the VLAN, Orion looks to see if the packet is destined (at the IP level) for a subnet which is associated with another locally resident L3 interface.

If there is no match at the IP destination subnet level, the packet is forwarded by default to the CPU for routing. However, an optional capability can be configured where-in such packets are L3 switched by the SOC to a default router address, for which a MAC address has been configured in the Default Router Table. This default router address can be global, or up to 9 defaults can be configured by destination subnet, with one of the defaults encompassing the "all others" case. These default routes can be modified by the CPU, but from the perspective of the Switch Fabric they are static.

If there is a match at the IP destination subnet level, then the Destination IP Address is searched in the L3 Table using IP Address as the key. If the IP address is not found then packet is given to the CPU for routing. If the IP Address match is found then the Mac Address of the next hop and the egress port number is picked up from this table.

In all cases, when the SOC performs L3 switching, it performs the following functions:

- validate IP checksum
- Substitution of the destination and Source MAC address
- Decrement TTL counter
- Re-calculate L3 CRC.
- Re-calculate the L2 CRC
- *These functions are only performed for IP packets with no options fields.*

Steps involved in Layer 3 switching:

1) Search ARL Table with Destination Mac address and check if the Mac Address is associated with an L3 interface.
2) Check if the Packet is an IP Packet (check for Ethernet V2 type, 802.3, tagged Ethernet V2 and Tagged 802.3 types of Packets). If the packet is not an IP Packet then send the Packet to the CPU for routing.
3) Check for the presence of Option Field in the packet. If Option fields are present then send the packet to CPU for routing.
4) Check for the Class D, also called Multicast Group IP Address. If the destination IP Address in the packet is a Multicast Group Address then send the Packet to the CPU for further processing.
5) Validate the IP Checksum.
6) Search the L3 Table with Destination IP Address as the key. If the entry is found then it will have the next Hop Mac Address, the egress port on which this packet has to be forwarded. If the Entry is not found then send the packet to CPU if no Default Router is configured (i.e Default Router is Empty). If Default Router is not empty then find a match in Default Router Table. This is done by ANDING the Destination IP Address with the Netmask in the Entry and checking if there is a match with the IP Address in the Entry. If there are multiple matches then one with highest Subnet Bitmap is selected. If the CPU Bit is set in that entry then a copy is send to the CPU (This is done so that the CPU can learn the new Route) and the Packet is modified before forwarding on to the destination port, as described below.
7) Decrement TTL, if it reaches zero then give it to CPU.

8) Recalculate IP Checksum, change Destination MAC Address with Next Hop Mac Address and Source Mac Address with Router Mac Address on the L3 Interface.
9) Check whether the packet should go out on the egress port as tagged or untagged and add or remove the Tagging Fields depending on this information.
10) Recalculate the L2 CRC.
11) Finally increment the Mib-2 interface counters.

Orion provides the following hooks to support L3 Switching.

1) L3 Table to do the Destination IP Address search. The table has following fields a) IP Address b) Next Hop Mac Address, c) the Egress port number and L3 interface Number.
2) Default Router Table.
3) Default Router Table Size.
4) L3 Interface Table to get the Router Mac Address and VLAN Id.
5) L3 Aging Timer.
6) ARL Logic which identifies the L3 Interface Address and starts the L3 Table search. The search key used is Destination IP Address. If the search is successful it decrements the TTL, recalculates IP checksum, changes the Destination and Source Mac Address, add or remove Tagging Fields depending on the egress Port and Vlan Id and recalculates the Ethernet Checksum.

L3 AGE_TIMER Register is used to set the L3_AGE_TIMER in seconds

L3_AGE_TIMER Configuration Register Format

30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0
Reserved						L3_AGE_TIMER									

Fields	# of Bits	Description
L3_AGE_TIMER	20	L3_AGE_TIMER – age Timer in seconds to age L3 Table Entries. Default is 300 seconds (range is from 10 sec to 1,000, 000 seconds)
Reserved	11	Reserved for future use.

L3 Table Format

30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0
IP Address															
Mac Addr 3				Mac Addr 2				Mac Addr 1				Mac Addr 0			
Res		L3H	L3 Interface Num			Port Number		Mac Addr 5				Mac Addr 4			

Fields	# of Bits	Description
IP Address	32	IP Address – is a 32 bit IP Address. The Destination IP Address in a packet is the used as a key in searching this table.
Mac Address	48	Mac Address is really the next Hop Mac Address. This Mac address is used as the Destination Mac Address in the forwarded IP Packet.
Port Number	5	Port Number – is the port number the packet has to go out if the Destination Mac Address matches this entry's IP Address.
L3 Interface Num	6	L3 Interface Num – This L3 Interface Number is used to get the Router Mac Address from the L3 Interface Table.
L3 Hit Bit	1	Hit bit – is used to check is there is hit on this Entry. The hit bit is set when the Source IP Address search matches this entry. The L3 Aging Process ages the entry if this bit is not set.
Reserved	4	Reserved for future use.

Default Router Table

If a match is not found in the L3 table for the Destination IP Address, then packet is forwarded to the default Router. Default Router Table contains Default Router Entries for each subnet. This table is just 9 entries deep and is similar to that of L3 table except that it also has netmask Information.

30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0
Subnet Address															
Mac Addr 3				Mac Addr 2				Mac Addr 1				Mac Addr 0			

Subnet Bits	L3 Interface Num	C	Port Number	Mac Addr 5	Mac Addr 4

Fields	# of Bits	Description
Subnet Address	32	Subnet Address – is a 32 bit IP Address of the Subnet.
Mac Address	48	Mac Address is really the next Hop Mac Address and in this case is the Mac Address of the default Router.
Port Number	5	Port Number – is the port number forwarded packet has to go out.
C Bit	1	C Bit – If this bit is send then send the packet to CPU also.
L3 Interface Num	6	L3 Interface Num – is L3 Interface Number.
Subnet Bits	5	Subnet Bits – is total number of Subnet Bits in the Subnet Mask. These bits are ANDED with Destination IP Address before comparing with Subnet Address.

Default Router Table Size Register

This is a 4 bit register which stores the number of valid entries in the Default Router Table.

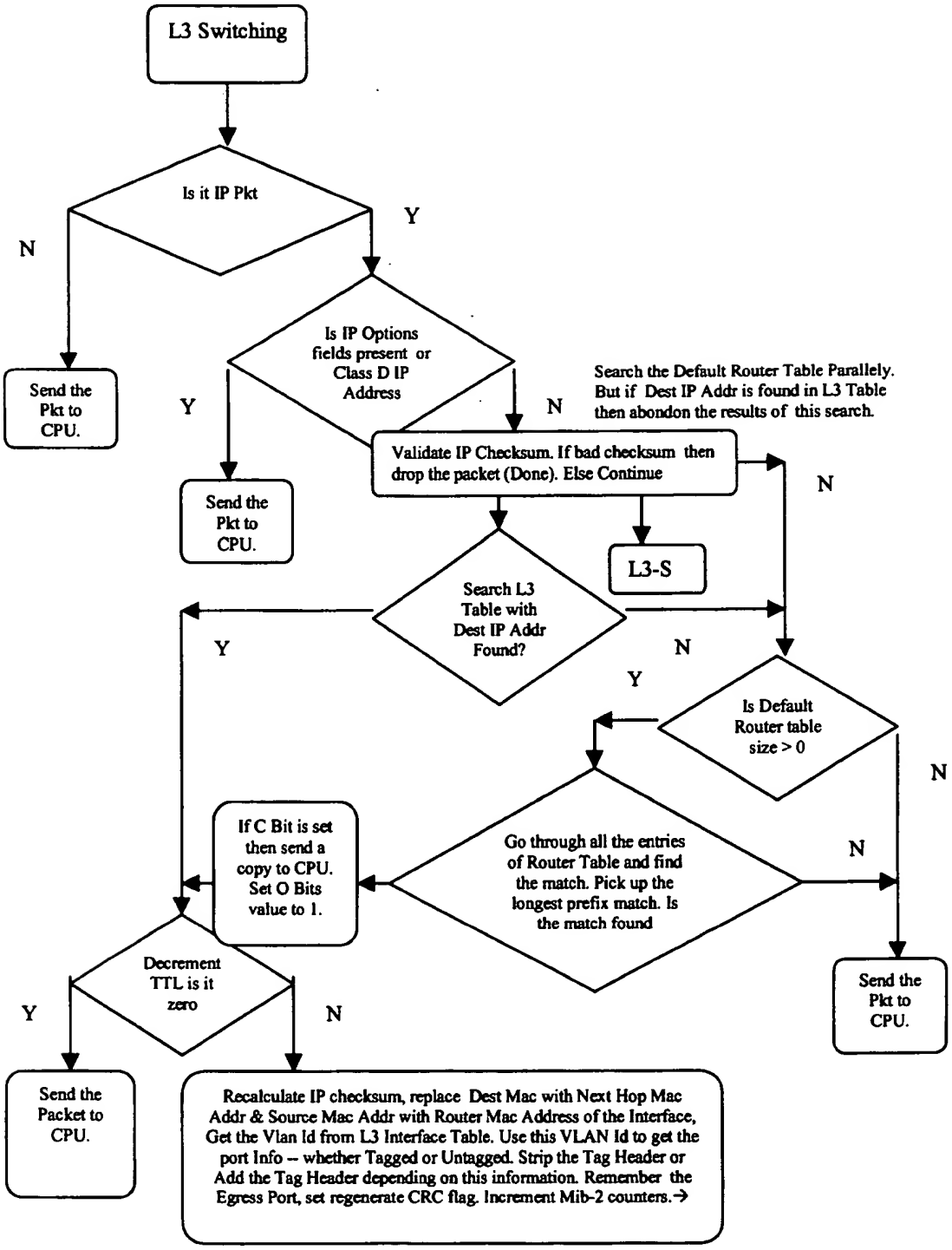
L3 Interface Table Format

This table is mainly used to get the Router Mac Address and Vlan Id from the L3 Interface Number. This table is 32 entries deep and 6 bytes wide. It is indexed by L3 Interface Number.

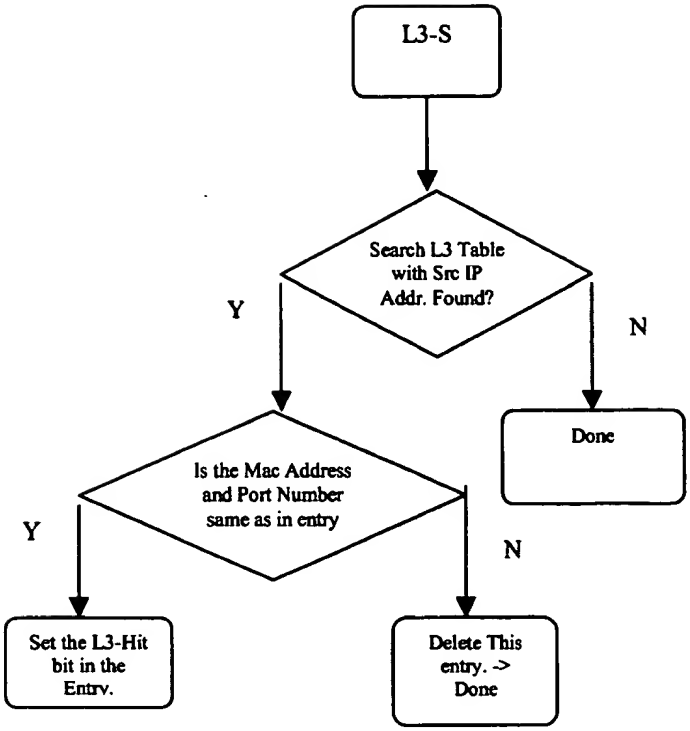
30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0
Router Mac Address (byte2..byte5)															
	Vlan Id							Router Mac Address (byte0 ..byte1)							

Fields	# of Bits	Description
Router Mac Address	48	Router Mac Address is really the L3 interface Mac address of the Router.
Vlan Id	12	Vlan Id – is the Vlan Id of this L3 interface. Vlan Id is used to get the information of egress port, - whether it is tagged or untagged.

L3 Switching Logic



L3-S – The Source IP Search need to be done for setting the Hit Bit. The IP packets Formats are given below.


L3-S
Search L3 Table with Src IP Addr. Found?
Y
N
Done
Is the Mac Address and Port Number same as in entry
Y
N
Set the L3-Hit bit in the Entry.
Delete This entry. -> Done

As discussed above, the SOC Architecture uses innovative technique for doing the Layer 3 Route Lookup. The Layer 3 Switching Logic uses Route Cache for End stations connected directly to one of the Layer 3 Interfaces of the switch and Default Router Table for the Stations that are not directly connected to one of the L3 interfaces. By using this technique, one can support large number of End Stations, which requires L3 switching by using relatively smaller Layer 3 Route Tables.

Applications of the Architecture

The primary application of this architecture is a High Performance Low Cost Layer 2 and Layer 3 Switch Fabric. This Switch fabric can be used to design for Workgroup, Power Workgroup, Desktop and Mid Tier Switches. It can also be used to design the Switching Blades of the High End Enterprise Backbone Switch. The low cost of the Switch Fabric brings the price per port of the Switches thus making it a very appealing solution for the end user.

Does not read well

Further Improvements

Further improvement of this Architecture include 1) Layer 4 Switching to optimize the load on the Servers, 2) providing interconnect capability so as to connect two or more SOCs thus enabling the Port Expansion Capability without sacrificing the Line Speed Switching capability between the ports.

What is claimed is:

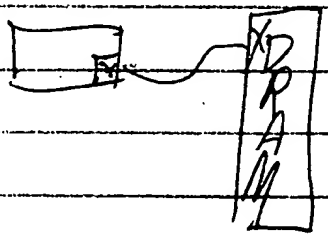
1. High Performance Low Cost Network Switching Architecture based on Distributed Hierarchical Shared Memory.
2. Dynamic Rerouting Algorithm for packet assembly in Global Buffer Pool. (Page 21)
3. Dynamic Buffer Allocation.
4. Maverick Networks Proprietary feedback driven Cell Channel.
5. State Machine driven programmable Rules Engine.
6. Policy Based Quality Of Service.
7. Load Balancing across trunk ports based on traffic classification.
8. Port Mirroring based on Programmable Filters.
9. Maverick Networks Innovative Layer 3 Switching implementation.

Competitors

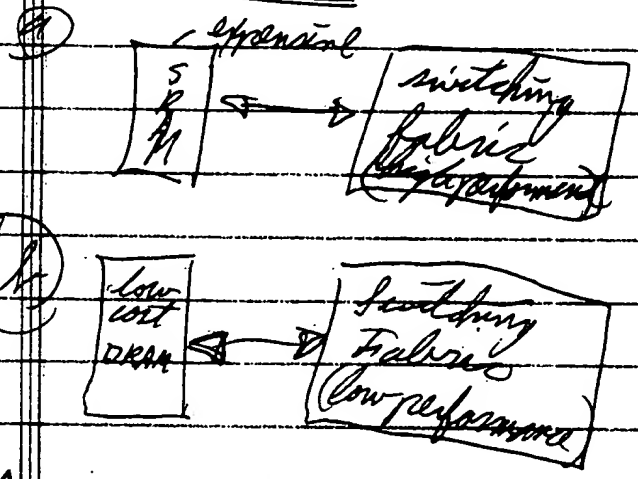
I-CUBE ← To DO

Galileo Technologies (public)
MMC Network (public)
PMC Sierra public
TI
Sony — 5,764,895
ACD

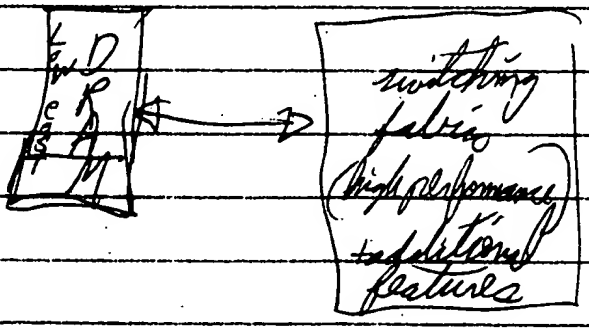
5,517,622
5,473,608
5,317,568



THEM



U.S



WWW. XYLAN. COM

THE SWITCH BOOK

OR

SWITCHING BOOK

Patent applications

(1.) Architectural setups - Switch Fabric
- A – use of low cost memory (DRAM) with Buffer Management scheme
- B – Self Balancing traffic flow
- C – filtering mechanism enabling packet application level filtering from layer 2 to layer 7

(2.) Classification of traffic based on the filtering mechanism

(3.) CPS Channel setups & functionality

We put a lot

DRAM - more

we take a performance hit

- refresh

- .

- page misses not a problem as in S-RAM

- cheaper + smaller area / cost per port

125 Hz - best case

what people look for

off chip can be 64MB - a significant

physical limitation

design would need more

address lines

packet aging

CBP - High speed memory

all based architectures

Ethernet - frame based technology

Cell makes it easier to handle

ATM is a cell based architecture

higher & higher integration

- put tables on chips

tables { ARL, Rules, L3, VLAN }

provides - performance - what you go off chip - you get performance hit

We have stat counters on the chips

Ingress & Egress functions
separated out
We have kept things modular in the architecture
e.g. put in a new interface

pipelining in RMM
- all ports are asynchronous
- we have to stitch cells together & also keep track of flow
- a consumer N buffers & break it up into N pipes
- each one does it to its own pacing

Tables

- longest prefix match
- in silicon, made assumptions on # stations supported
  - route cache - 32 bit IPv4
  - default router lookup
    - longest prefix match
    - splitting into 2 tables, we cannot get best of both

worlds.

- expensive
- complex
-

L.P.M.

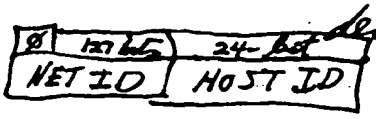
IP address - 32 bits

key

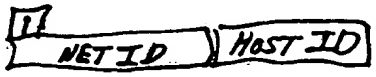
aa.bb.cc.dd

192.168.24.1

see page 44

Class - A - 

divisible

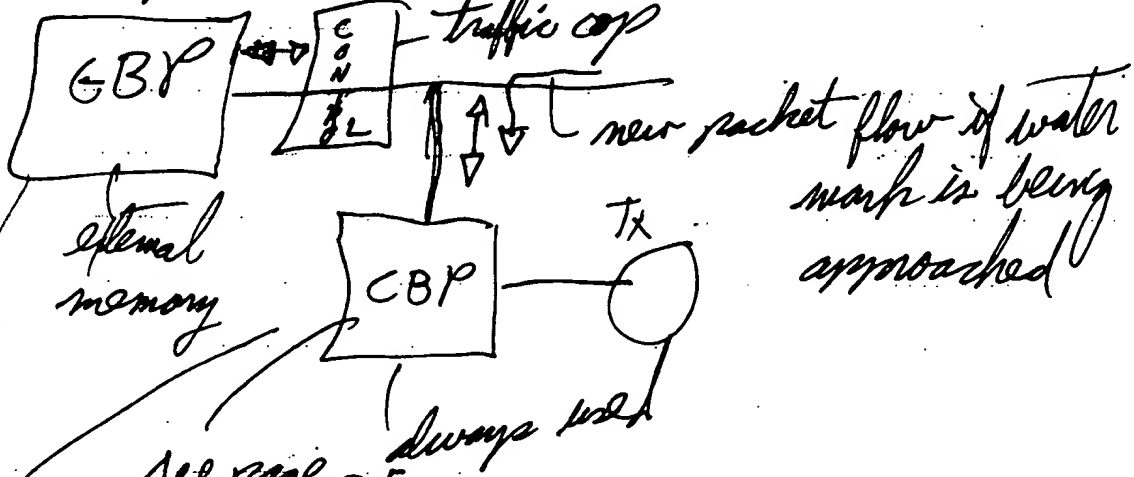
Class - B - 

self Balanced architecture

watchpoint - off chip memory

↳ closest point

data flow



see page 25

everything will go in one or the other

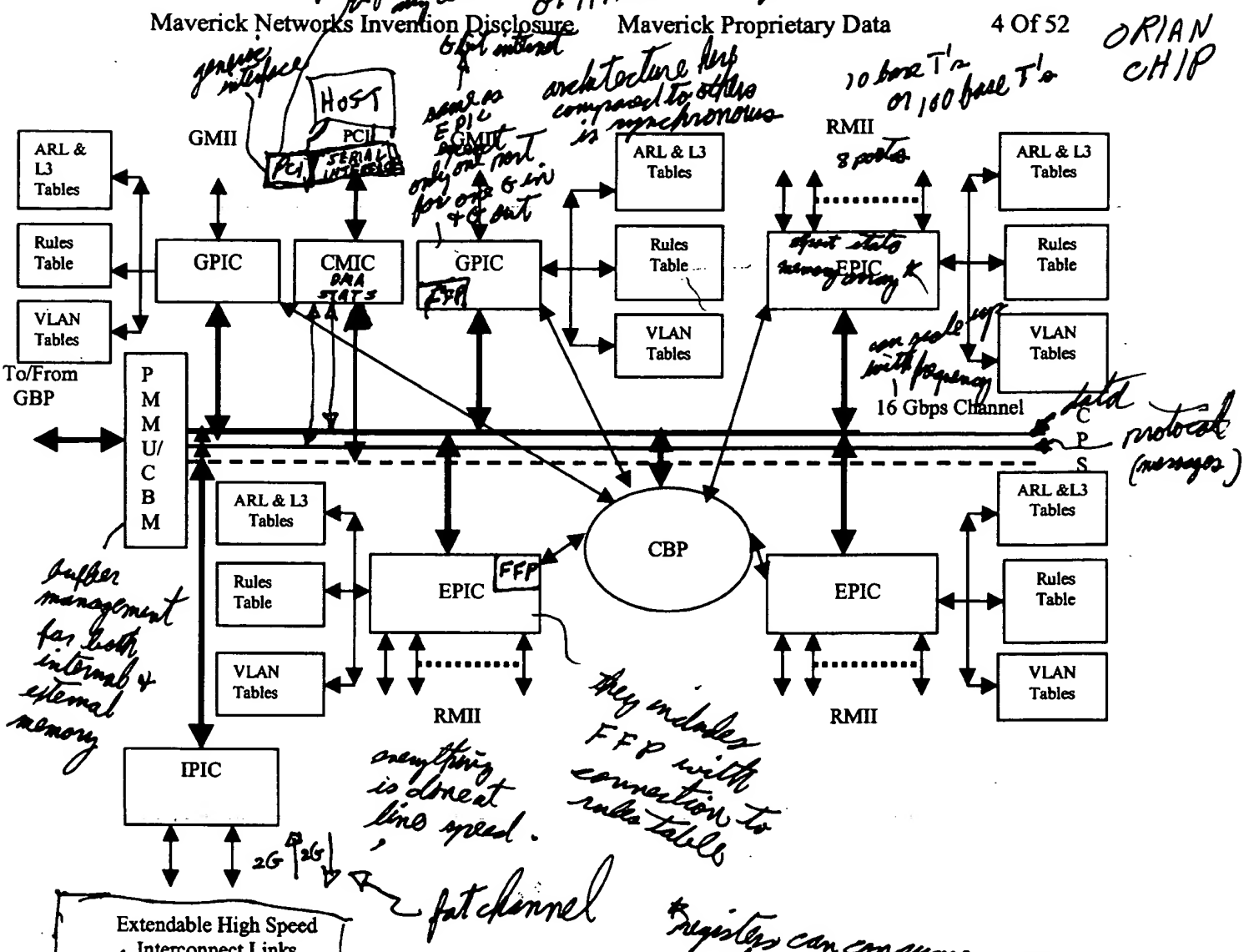


Figure 1: SOC Architectural Block Diagram

The following are the major blocks of SOC:

- Ethernet Port Interface Controller (EPIC)
- Gigabit Port Interface Controller (GPIC)
- Interconnect Port Interface Controller (IPIC)
- CPU Management Interface Controller (CMIC)
- Common Buffer Pool (CBP) / Common Buffer Manager (CBM)
- Global Buffer Pool (GBP)

Witnessed and Understood By:	Date:
Submitter(s) Signature(s):	Date(s):